Environmental Data Logging

Developers:

Alec Wassill, Bryan Martinez, Cameron McNaught, Josh Lynn

Client:

James Ravenhill

Instructors:

Dave Watson, John O'Loughlin

Table of Contents

Table of Figures

Preface

The purpose of this document is to identify and outline the requirements and functionality that the application must meet in order to accomplish the client's needs.

Document Standards

The following standards will outline how content is displayed in this document:

Keywords and key phrases will be italicized. All of these keywords and key phrases can be found in the *Glossary* section of the document, except for any keywords found within the *Data Dictionary* portion of this document. Those keywords will be italicized to indicate the keyword can be found in a separate portion of the *Data Dictionary*.

Words and phrases that indicate sub-sections of a main section and do not need to be indexed will be shown as bold text.

Acronyms will have a superscript number immediately proceeding the acronym. These numbers correspond to the relative number at the bottom of the page stating the expanded term. These terms are also expanded and described in more detail in the *Glossary* section of this document.

Primary headings will use Calibri font, size 16 and light blue (hexadecimal code: # 2E74B5). The heading will also be centered.

Secondary headings are Calibri, size 13, light blue (hexadecimal code: # 2E74B5), left-aligned font with no indentation.

Tertiary headings are Calibri, size 12, dark blue (hexadecimal code: # 1F4D78), left-aligned font with 0.13" indentation.

Quaternary headings are Calibri, size 11, light blue (hexadecimal code: # 2E74B5), italicized, left-aligned font with 0.19" indentation.

Quinary headings are Calibri, size 11, light blue (hexadecimal code: # 2E74B5), left-aligned font with 0.25" indentation.

The main text in the document will be Calibri, size 11 font and justified with 8pt spacing proceeding the body of text. All text will be indented 0.06" on the left further than that of the heading preceding it or in the case to produce an effect of hierarchy in the document without the use of bulleted or numbered lists, headings or change in font style (bold); as this section showcases.

Commas will be used using that AP standard; not using the Oxford/Serial style.

All references will be cited using IEEE standards.

Every page, with exempt to the title page, will include the heading *Requirements Document* in the top right corner. As well each page will include the name of the project in the bottom left corner, the page number at the bottom center and the current or the date the document was printed on, in the bottom right corner of the page.

No section, table, paragraph, sentence, or text should start on one page and continue onto subsequent pages, unless absolutely necessary. Should a break need to be inserted to follow this rule then a section break will be used; not a page break.

All diagrams will be presented in the best possible manner that suits the shape and size of that diagram. As long as the diagram, in combination with it's associated heading and caption, does not conflict with the paragraph directly above this one.

All captions to diagrams and pictures will appear at the bottom of the page in portrait orientation; even if the diagram or picture is in landscape orientation.

All tables used will alternate shading color every other row and have appropriate borders for that table. The only exception to this is the *Glossary* section which will have no shading on any cells and no border.

Bullets will use the ➢ symbol (Wingdings: 216) for all primary bullet levels. Secondary bullet levels will use the ❖ (Wingdings: 118).

Page numbers will use Standard English Arabic numbering starting with the *Preface* section, all preceding pages, with the exception of the title page, will use lowercase Roman Numerals.

In the digital copy of the document, all headings listed in the table of contents as well as the table of figures will be links to the associated heading or figure being clicked on.

Any paragraphs, tables or lists within a section that require additional spacing beyond that of it's preassigned spacing can use 1 line of blank space with single spacing and no spacing preceding or proceeding the blank space.

The title text and any references to specific code will use Papyrus font style. Further font style will depend on the text's position in the document in accordance with previously mentioned rules.

User Requirements

Business Overview and Objectives

The client, James Ravenhill is a researcher working for the *Environmental Technologies* division of *ARIS*[1]. The Environmental Technologies lab takes various samples (water, soil, air) and run tests on these samples to uncover the chemicals inside. Through various machines, the client takes the data results from these tests and generates a report on the machine's findings.

The *Environmental Technologies* research team leads applied research in *Clean Technologies* associated with energy and the environment. The *Environmental Technologies* research team develops strategies and solutions which includes the reduction of harmful impacts to air, soil, and water. The research is grounded in science, technology and engineering and focuses on technologies from bench to pilot scale in order to address today's energy and environmental challenges. The *Environmental Technologies* research advances a body of highly skilled knowledge and technological know-how to benefit industry and society, while providing experimental learning to students from *SAIT's* [2]applied science and technology disciplines. This research facilitates technology integration with regulatory compliances, speeds innovation and helps accelerate innovative technology transfer to serve the nation and the world in the 21st century. [1]

Project Overview

The *Environmental Technologies* team collects a lot of data when conducting research projects. Many of the pieces of equipment in their research lab are able to output data to a computer through USB and Ethernet connections. As well some devices have *APIs* [3]that allow real-time data collection.

The *Environmental Technologies* team would like a web-based visualization system that allows the data, collected from lab work, to be archived and displayed via an internet browser.

[1] Applied Research and Innovation Services
[2] Southern Alberta Institute of Technology
[3] Application Programming Interface

Project Scope

The *Environmental Technologies Lab* is in need of a system that can import data from multiple devices into a centralized database that has a web based user interface to access and generate reports.

Features of the application include:

- Automate the transmission of a *CSV*[4] file to the database from multiple devices
- If the device is not connected to a computer, the application will provide the user of the application the ability to manually input the data into the application
- Allowing users the ability to login, view and input data and generate reports simultaneously
- Administrators will be able to add and modify user permissions and device information
- Retrieve and display data from the database

The system is required to import a CSV file with the sample data from each device into a centralized database. These devices can run on a Windows operating system and operate with or without a network connection. If the device can not generate a CSV file, the lab technician will have the ability to manually input data using the web application.

System Environment

The system will primarily be used in a chemistry lab with 12 computers and various devices. The system will combine many data sheets into a centralized database that can be easily accessed from the web. This will eliminate the redundancy of having many versions of the data sheets distributed.

Current System

Currently the members of the *Environment Technologies lab* are using various devices to capture data from their samples and are either manually entering the data into Excel spreadsheets or having the devices automatically generate a CSV file and then storing it locally on computers or devices within the lab.

To generate reports our client obtains the data from the devices and manually compiles all the data to generate tables and graphs within Microsoft Excel. The current system of manually gathering data and generating reports takes approximately 40 hours per report.

[4] Comma Separated Values

System Requirements

Fact-Finding Methodology

Information on how to approach the problem was primarily discovered through the initial interview with the client. Some of the questions posed by the development team include:

- **Do we need the ability to add and remove devices?**
  Yes. Each user should have the ability to add and remove devices as needed.
- **Is the data input coming solely from the devices or will the user need to be able to input data directly into the application?**
  Some of the devices have no means to connect to a computer and as such will need to have their data manually inputted into the application.
- **What operating system and browser will the application be run on?**
  All the computers run Windows 7 and the browser isn't specified.
- **Is there currently a database where the information is stored?**
  No. Currently all the data for each device is stored in CSV files on different computers. When it comes time to gather this data it is transferred to my, the client's, computer where I load all the files into a single Excel document to generate my reports.
- **Does this application need to run on a mobile device?**
  Does a laptop count? The main application will run on desktop computers, however it may need to be run on laptop or even sometimes tablets.
- **How many types of users are going to use the system?**
  There will only need to be 2 types of users. I would like a standard user that has the ability to do everything and an administrative user who has the ability to add and remove users from the system.

Use Case Diagram

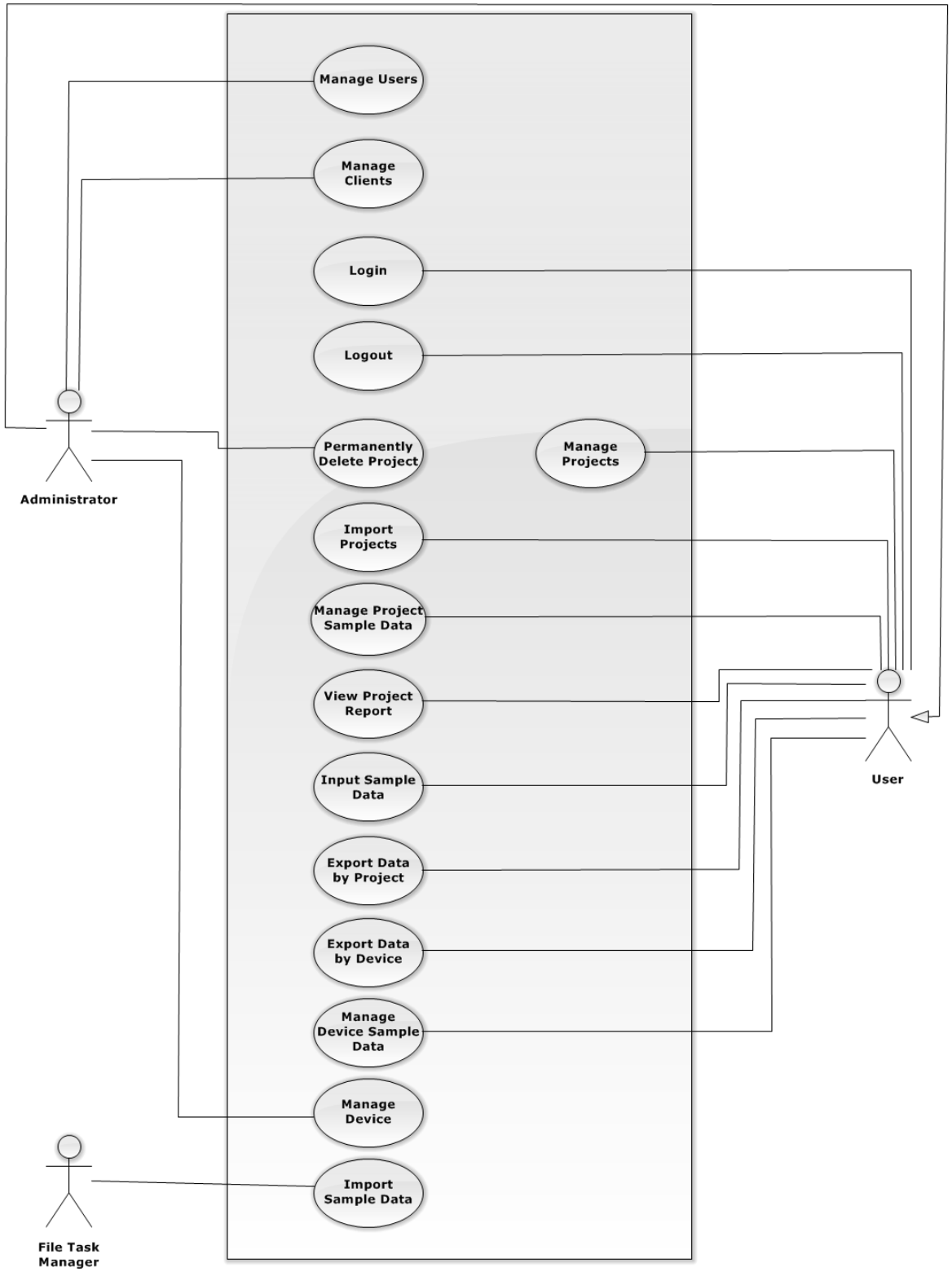

Figure 1 - Use Case Diagram

Use Case Descriptions and Scenarios

Description Summary

- **Manage Users**
  The administrator will have the ability to add and update user's information.
- **Manage Clients**
  The administrator will have the ability to add and update client information.
- **Login**
  The user will access the application.
- **Logout**
  The user will leave the application.
- **Manage Projects**
  The user will be able to view all projects, add and update projects.
- **Permanently Delete Projects**
  The administrator will permanently delete the project from the application.
- **Import Projects**
  The user can import existing projects and the associated data from a data file outside of the application.
- **Manage Project Sample Data**
  The administrator has the ability to manage data for a project.
- **View Project Report**
  The user can view a printable report containing all visible data for that project.
- **Input Sample Data**
  The user can manually enter information, by device, for the selected project.
- **Export Data by Project**
  The user can export the data for a selected project to a CSV file.
- **Export Data by Device**
  The user can export the data for a selected device to a CSV file.
- **Manage Device Data**
  The administrator has the ability to manage data for a device.
- **Manage Device**
  The user will have the ability to view all devices and update the device information.
- **Import Sample Data**
  The device service will automatically import new data from the device into the database.

Use Case Description Details

Manage Users Extended Use Case Description

Use Case Name	Manage Users
Actor	Administrator
Description	The administrator will have the ability to add and update a user's information
Pre-conditions	➢ The actor who is logged must have administrator level permissions ➢ The actor selects the *Manage Users* button from the navigation bar
Scenario	1. The actor will be able to view table showing all users 2. The actor will have the ability to add and update user information from this screen 3. Add User 3.1. The actor will click the *Add User* button 3.2. A pop-up box will appear for the actor to enter new user's information including: 3.2.1. Full name 3.2.2. Email 3.2.3. Password 3.2.4. If the user has administrator level permissions? 3.3. The actor can choose to save the new user to application by clicking on the *Save* button 3.4. The actor can choose to not save the new user to the application by clicking on the *Cancel* button 4. Update User 4.1. The actor will click a user in the table 4.2. A button will appear allowing the actor to update the selected user's information 4.3. When the actor clicks on the *Update* button a pop-up box will appear allowing the actor to update the selected user's: 4.3.1. Permission status 4.3.2. Full Name 4.3.3. Password 4.3.4. Email 4.3.5. Status
Post-conditions	➢ No post-conditions exist for this use case

Manage Clients Extended Use Case Description

Use Case Name	Manage Clients
Actor Involved	Administrator
Description	The administrator will have the ability to add and update a client's information
Pre-conditions	➢ The actor who is logged must have administrator level permissions ➢ The actor selects the *Manage Clients* button from the navigation bar
Scenario	1. The actor will be able to view table showing all clients 2. The actor will have the ability to add, delete and modify client's information from this screen 3. Add Client 3.1. The actor will click the *Add Client* button 3.2. A pop-up box will appear for the actor to enter new client's information including: 3.2.1. Company Name 3.2.2. Company representative/contact 3.2.3. Company phone number 3.2.4. Company contact email 3.3. The actor can choose to save the new client to application by clicking on the *Save* button 3.4. The actor can choose to not save the new client to the application by clicking on the *Cancel* button 4. Update Client 4.1. The actor will click a client in the table 4.2. A button will appear allowing the actor to update the selected client's information 4.3. When the actor clicks on the *Update* button a pop-up box will appear allowing the actor to update the selected client's: 4.3.1. Company Name 4.3.2. Company representative/contact 4.3.3. Company phone number 4.3.4. Company contact email
Post-conditions	➢ No post-conditions exist for this use case

Login Extended Use Case Description

Use Case Name	Login
Actors Involved	User, Administrator
Description	The actor will be able to access the application
Pre-conditions	➢ The actor navigates to the application URL in their internet browser
Scenario	1. The actor will enter their email address in the textbox under the text *Enter Email* 2. The actor will enter their password in the textbox under the text *Enter Password* 3. The actor will click on the *Login* button 4. If the email and password combination that the actor entered into the application is invalid the actor will be allowed to re-enter their login credentials
Post-conditions	➢ If the actor provides the correct login credentials they will be redirected to the application home page

Logout Extended Use Case Description

Use Case Name	Logout
Actors Involved	Administrator, User
Description	The user will leave the application
Pre-conditions	➢ The actor must be currently logged into the application
Scenario	1. The actor will click on the *Logout* button
Post-conditions	➢ The actor is redirected to the login screen

Manage Projects Extended Use Case Description

Use Case Name	Manage Projects
Actors Involved	Administrator, User
Description	The user will be able to view all projects, add and update projects
Pre-conditions	➢ The actor clicks on the *Projects* button from the navigation bar
Scenario	1. The actor will be able to view table showing all projects 2. The actor will have the ability to add and update project information from this screen 3. Add Project 3.1. The actor will click the *Add Project* button 3.2. A pop-up box will appear for the actor to enter new project's information including: 3.2.1. Name 3.2.2. Start date and time 3.2.3. Client 3.2.4. Any additional comments 3.3. The actor can choose to save the new project to application by clicking on the *Save* button 3.4. The actor can choose to not save the new project to the application by clicking on the *Cancel* button 4. Update Project 4.1. The actor will click a project in the table 4.2. A button will appear allowing the actor to update the selected project's information 4.3. When the actor clicks on the *Update* button a pop-up box will appear allowing the actor to update the selected client's: 4.3.1. Name 4.3.2. Start date and time 4.3.3. If project completed 4.3.4. Client 4.3.5. Any additional comments
Post-conditions	➢ No post-condition exists for this use case

Permanently Delete Project Extended Use Case Description

Use Case Name	Permanently Delete Project
Actor Involved	Administrator
Description	The administrator will permanently delete the project from the application
Pre-conditions	➢ The actor clicks on the *Projects* button from the *Admin* sub-navigation bar
Scenario	1. The actor will be able to view table showing all projects 2. The actor will click on a project in the table 3. A button will appear allowing the actor to delete the project 4. When the actor clicks on the *Delete* button a confirmation pop-up will appear confirming the actor wants to delete the project 4.1. If actor clicks on the *Confirm* button the application will delete the project and all associated data for that project 4.2. If the actor click on the *Cancel* button the actor will return to the view projects screen
Post-conditions	➢ The project and all associated data is permanently deleted from database

Import Project Extended Use Case Description

Use Case Name	Import Projects
Actors Involved	Administrator, User
Description	The user can import existing projects and the associated data from a data file outside of the application
Pre-conditions	➢ The actor clicks on the *Projects* button from the navigation bar
Scenario	1. The actor clicks on the *Import Projects* button above the table showing all projects 2. A pop-up box will appear asking the actor to select the CSV file to be imported into the application 3. After the actor selects the CSV file to be imported, the actor will click on the *Import Projects* button
Post-conditions	➢ The CSV file will be sent to the application to be processed and added to the database ➢ The actor will return to the view projects screen ➢ If the import is successful a notification will appear informing the actor the data has been added to the system with the option to refresh the page ➢ If the import is unsuccessful a notification will appear informing the actor the import was unsuccessful and suggesting to retry the import

Manage Project Sample Data Extended Use Case Description

Use Case Name	Manage Project Sample Data
Actors Involved	Administrator, User
Description	The administrator has the ability to hide data for an project
Pre-conditions	1. The actor clicks on the Projects button in the navigation bar 2. The actor selects a project from the table of projects 3. A button appears allowing the actor to view project sample data 4. The actor click on the *Project Data* button 5. The actor is redirected to the view project sample data screen
Scenario	1. The actor is able to view all sample data associated to the project selected 2. The actor is able to hide sample for the project 2.1. The actor clicks on a line of data in the table 2.2. The actor checks or unchecks the *Hidden* checkbox on the same line as the selected row
Post-conditions	➢ If the actor checks the checkbox, the sample data will remain in the application, but will not be seen on other pages when the sample data for that project is displayed ➢ If the actor unchecks the checkbox, the sample data be seen on other pages when the sample data for that project is displayed

View Project Report Extended Use Case Description

Use Case Name	View Project Report
Actors Involved	Administrator, User
Description	The user can view a printable report containing all visible data for that project
Pre-conditions	➢ The actor clicks on the *Projects* button in the navigation bar ➢ The actor is redirected to the view project screen
Scenario	1. The actor is able to view all projects 2. The actor clicks on a project in the table 3. A button appears allowing the actor to view a report for that project 4. The actor clicks on the *View Report* button 4.1. A pop-up box will appear for the actor to select the report type from a list of pre-defined report templates 4.2. The actor clicks on the *View Report* button
Post-conditions	➢ The application opens a printable PDF copy of the report in a new window with all the data for the selected project ➢ The actor is returned to the view projects screen

Input Sample Data Extended Use Case Description

Use Case Name	Input Sample Data
Actors Involved	Administrator, User
Description	The actor can manually enter information, by device, for the selected project
Pre-condition 1	1. The actor clicks on the *Projects* button in the navigation bar 2. The actor selects a project from the table 3. A button appears allowing the user to input data for the selected project 4. The actor clicks on the *Input Data* button
Pre-condition 2	1. The actor clicks on the *Devices* button in the navigation bar 2. The actor selects a device from the table 3. A button appears allowing the user to input data for the selected device 4. The actor clicks on the *Input Data* button
Pre-condition 3	1. The actor clicks on the *Samples* button in the navigation bar 2. The actor clicks on the *New Sample* button
Scenario 1	1. The actor selects the device they are entering information for from a dropdown menu 2. The actor enters data into all the required fields 3. When all of the required fields are filled out the actor clicks on the *Save* button 4. At any point the actor can click on the *Cancel* button
Scenario 2	1. The actor selects the project they are entering information for from a dropdown menu 2. The actor enters data into all the required fields 3. When all of the required fields are filled out the actor clicks on the *Save* button 4. At any point the actor can click on the *Cancel* button
Scenario 3	1. The actor selects the project they are entering information for from a dropdown menu 2. The actor selects the device they are entering information for from a dropdown menu 3. The actor enters data into all the required fields 4. When all of the required fields are filled out the actor clicks on the *Save* button 5. At any point the actor can click on the *Cancel* button
Post-conditions	➢ If the actor clicks on the *Save* button, the application will save the data entered to the database. The actor will be redirected back to the previous screen ➢ If the actor click on the *Cancel* button all data entered will be discarded and the actor is redirected back to the previous screen

Export Data by Project Extended Use Case Description

Use Case Name	Export Data by Project
Actors Involved	Administrator, User
Description	The actor can export the data for a selected project to a CSV file
Pre-conditions	1. The actor clicks on the *Projects* button in the navigation bar 2. The actor clicks on a project in the table
Scenario	1. A button will appear allowing the actor to export the data for the selected project to a CSV file 2. The actor will click on the *Export* button 3. A pop-up box will appear asking the actor where the file should be saved to 4. The actor will select a folder from the directory hierarchy for where the file will be saved to 5. The actor will click on the *Save* button 6. At any point the actor can click on the *Cancel* button
Post-conditions	➢ If the actor clicks on the *Save* button, the application will create and save a CSV file with all the information for the selected project to the specified folder. The actor will be redirected back to the *Projects* screen ➢ If the actor clicks on the *Cancel* button, the application will redirect the actor to the *Projects* screen

Export Data by Device Extended Use Case Description

Use Case Name	Export Data by Device
Actors Involved	Administrator, User
Description	The actor can export the data for a selected device to a CSV file
Pre-conditions	3. The actor clicks on the *Devices* button in the navigation bar 4. The actor clicks on a device in the table
Scenario	7. A button will appear allowing the actor to export the data for the selected device to a CSV file 8. The actor will click on the *Export* button 9. A pop-up box will appear asking the actor where the file should be saved to 10. The actor will select a folder from the directory hierarchy for where the file will be saved to 11. The actor will click on the *Save* button 12. At any point the actor can click on the *Cancel* button
Post-conditions	➢ If the actor clicks on the *Save* button, the application will create and save a CSV file with all the information for the selected device to the specified folder. The actor will be redirected back to the *Devices* screen ➢ If the actor clicks on the *Cancel* button, the application will redirect the actor to the *Devices* screen

Manage Device Data Extended Use Case Description

Use Case Name	Manage Device Sample Data
Actors Involved	Administrator, User
Description	The actor has the ability to hide data for an device
Pre-conditions	1. The actor clicks on the *Device* button in the navigation bar 2. The actor selects a device from the table 3. A button will appear allowing the user to manage the device data 4. The actor will click on the *Edit* button 5. The actor will be redirected to a view device data screen
Scenario	1. The actor will click on a data value and be able to edit the value 2. The actor will have the ability to hide/unhide device data 2.1. The actor will click on the device in the table 2.2. The actor will check or uncheck the *Visible* checkbox on the selected line 3. The actor will click on the *Save* button 4. At any point the actor can click on the *Cancel* button
Post-conditions	➢ If the actor clicks on the *Save* button the application will save all changes made to the database. The actor will be redirected to the *Devices* screen ➢ If the actor clicks on the *Cancel* button the actor will be redirected to the *Devices* screen

Manage Devices Extended Use Case Description

Use Case Name	Manage Devices
Actor Involved	Administrator, User
Description	The user will have the ability to view all devices and update the device information
Pre-conditions	➢ The actor must be logged in
Scenario	1. The actor selects the *Devices* button form the navigation bar 2. The actor select a device from the view table showing all devices 3. The actor clicks on the *Edit* button 4. The application will navigate to a new screen to allow the actor to edit the device information 5. The actor will be able to update the following information for the device: 5.1. Name 5.2. Location 5.3. Status 6. The actor can choose to save the updated changes or cancel the changes
Post-conditions	➢ If the actor clicks on the *Save* button, all changes made to the device information will be saved to the database ➢ If the actor clicks on the *Cancel* button, all changes made to the device information will not be saved ➢ The application will navigate the actor back to the view table showing all devices

Import Sample Data Extended Use Case Description

Use Case Name	Import Sample Data
Actor Involved	Device service
Description	The device service will automatically import new data from the device into the application
Pre-conditions	No pre-conditions exists for this use case
Scenario	1. The actor adds a new CSV file to the application automatic import folder 2. The application will read the CSV file and extract the data 3. The application will save the data extracted to the database
Post-conditions	No post-conditions exist for this use case

Non-Functional Requirements

Other functions that the application will have to adhere to include:

- Application should take no longer than 10 seconds to generate reports
- Application should take no longer than 10 seconds to generate CSV files
- Application should be accessible on any computer in the lab
- A new user should be able to log into the system within 1 minute of being added to the application by an administrator
- Immediately after a user's status is set to false, the selected user will not be able to log on
- When adding or updating a user account, the username must be a valid email address
- For non-network connected devices, after the device is added to the system any user can manually input data for the device
- For network connected devices, after the device is added to the system, any user can start associating device data to projects
- The process of deleting a device will complete within 1 minute
- Changes when editing a device can be seen by all users within 1 minute of changes being saved
- Database will have enough storage space to hold all information
- When viewing data, the table displaying the data will take no longer than 10 seconds to populate with data
- When viewing data, the table displaying the data will apply user specified filters and re-populate the tables within 10 seconds
- When editing data, changes will be visible to all users within 5 seconds of saving changes
- Reports will be generated and displayed within 30 seconds
- Exporting data will take no longer than 1 minute
- Service to automatically send data from device to application must be running at all times

System Interface Requirements

The application will be able to interface will multiple devices; some of these devices will be accessed through their built in API's to access and retrieve data. Other devices for which the application is unable to access the device API will interface with the computers that are connected to those devices.

As well the application may need the ability to interface with the *Environmental Technologies* lab or SAIT's database.

Maintainability & Administrator Requirements

The application administrator will have the ability to add and remove standard users as needed. As well the administrator will have the ability to modify any user's permissions. These levels include who is allowed to view and edit raw data, who can generate reports, who can modify save file locations, who can view reports, etc.

The application will be maintained through using this document, various diagrams and well documented code.

Usability Requirements

The application will primarily be used by the lab technician's in the *Environmental Technologies* lab. The application will need to be visually simple in its design while having a wide range of functionality. Some of these functionalities include:

- Login to application
- Add and Update users
- View and Edit data
- View and generate reports depending on project
- Manage account settings

Problem Domain Package Diagram

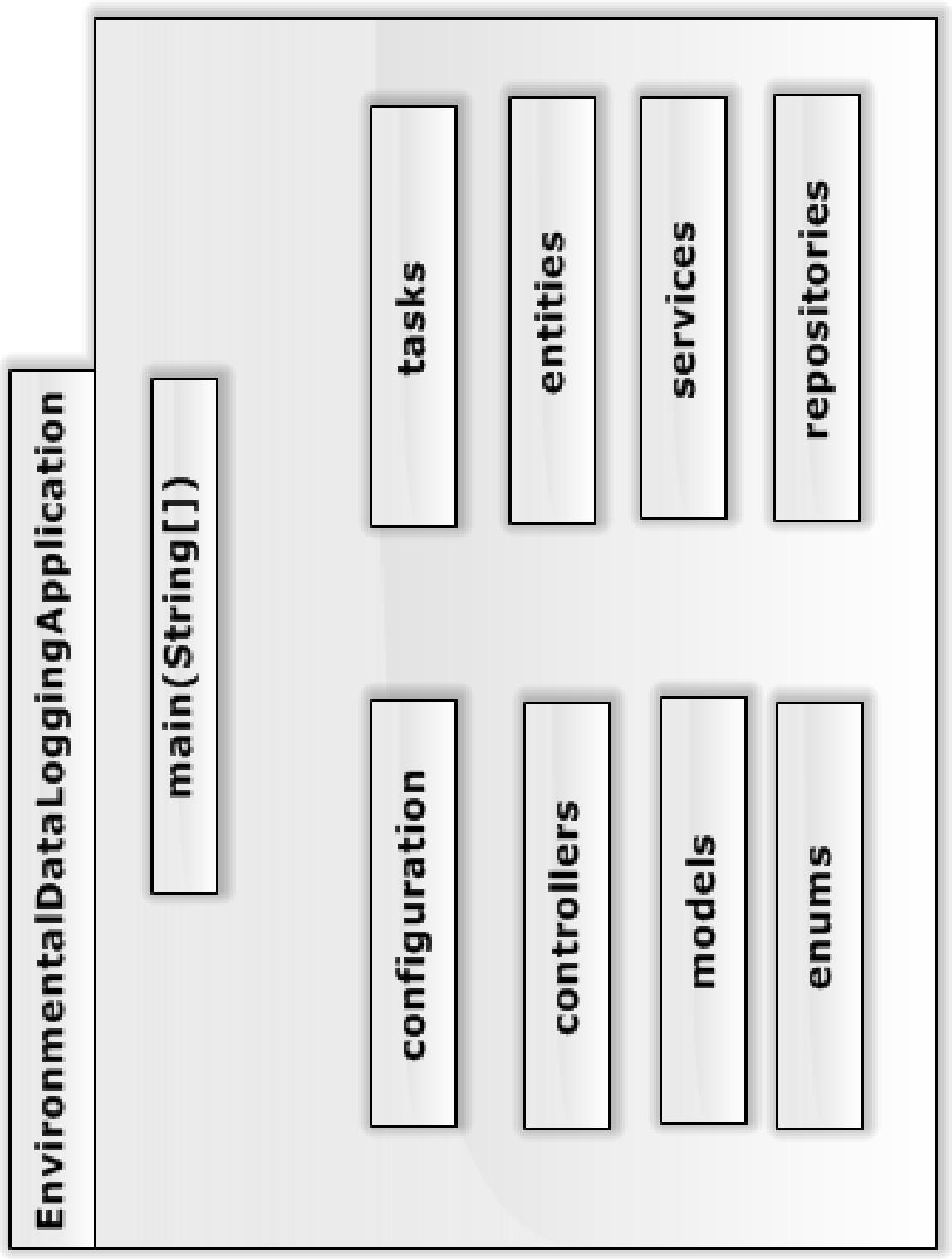

Figure 2 - Package Diagram

Problem Domain Class Diagram

Configuration Class Diagram

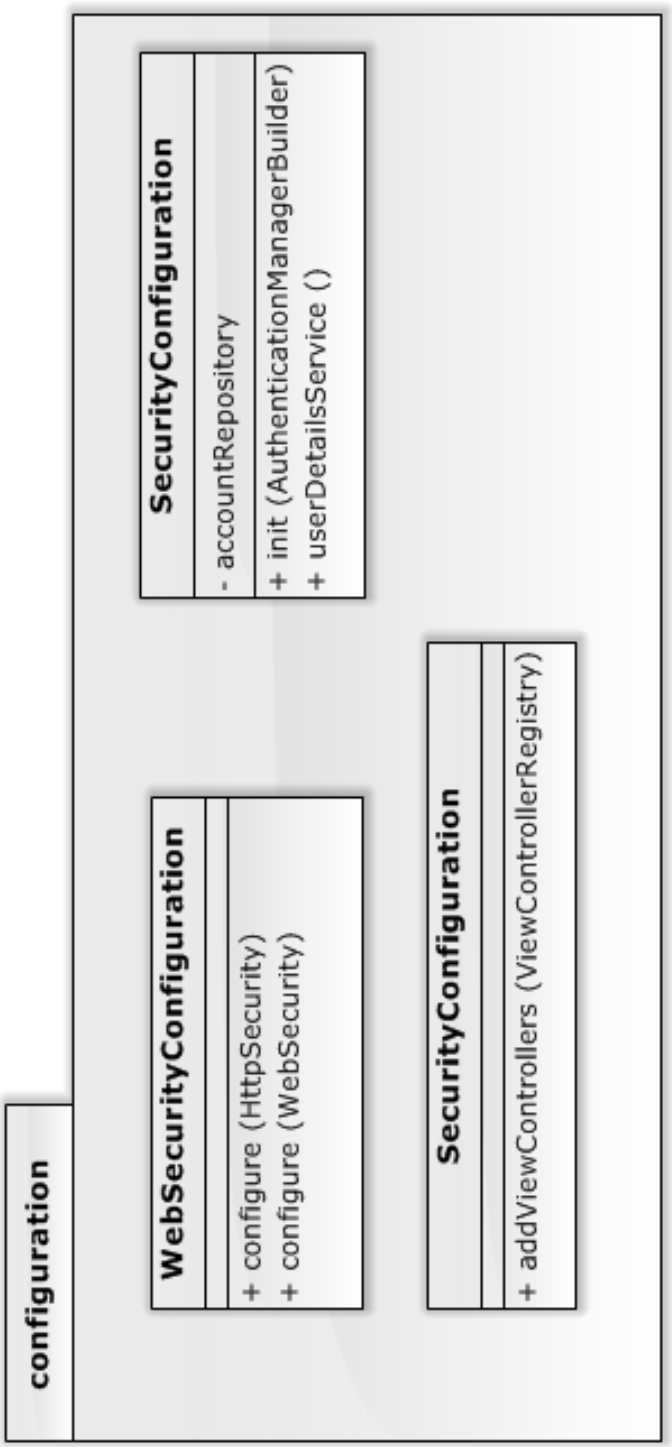

Figure 3 - Configuration Class Diagram

Controller Class Diagram

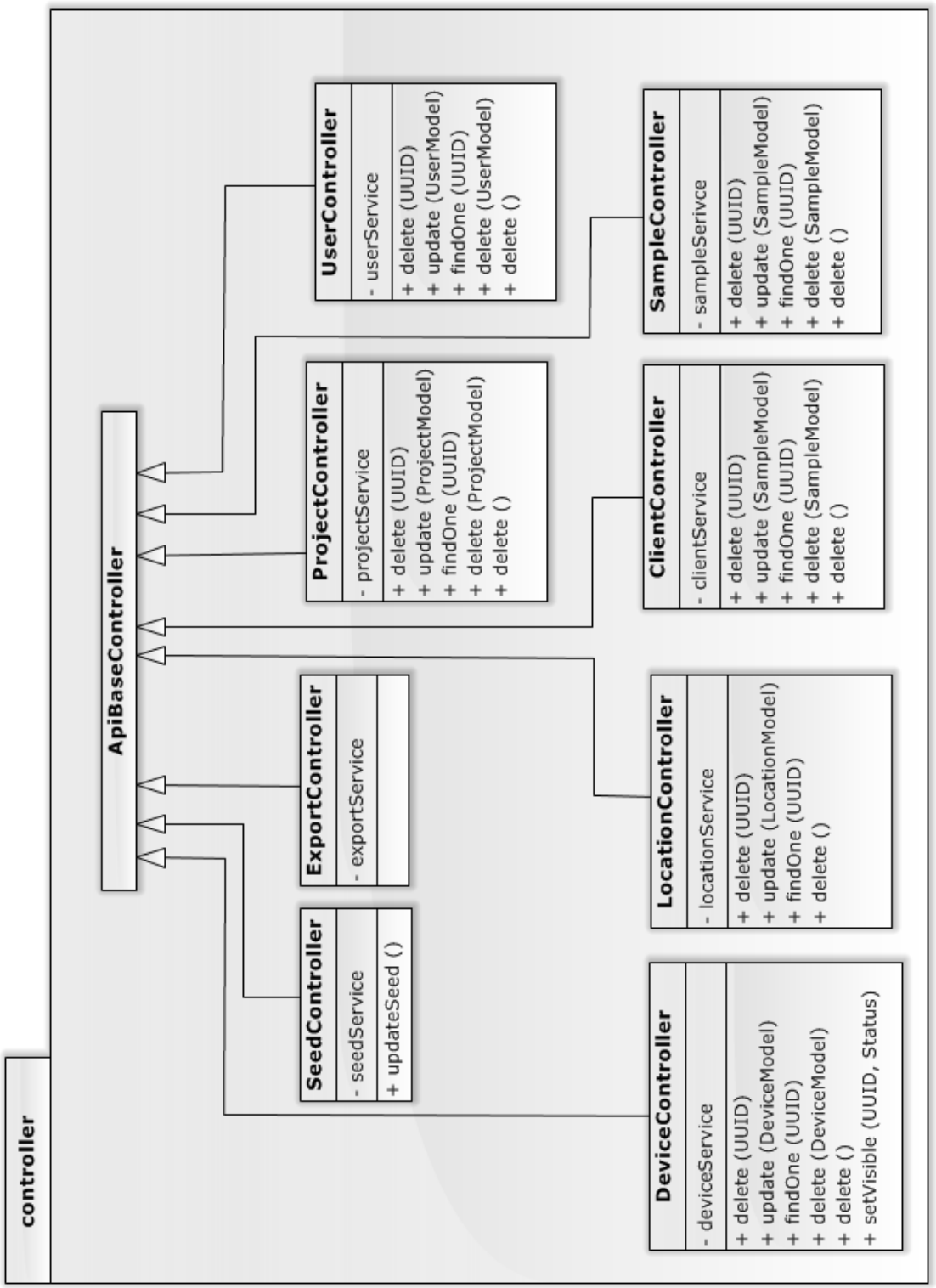

Figure 4 - Controller Class Diagram

Entity Class Diagram

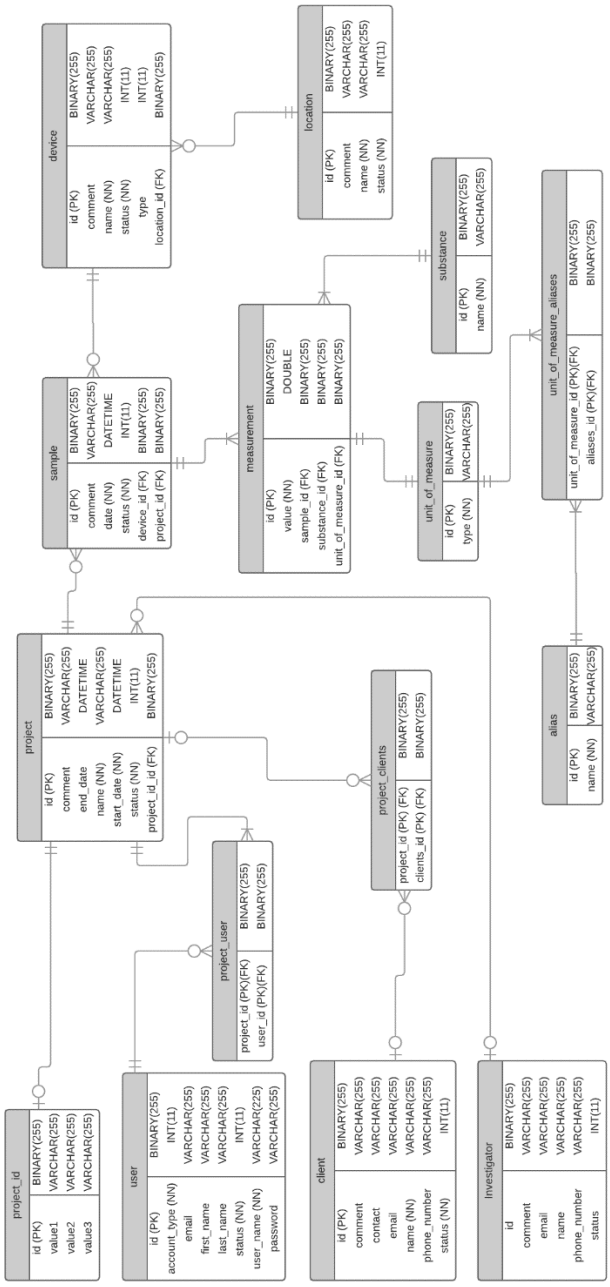

Figure 5 - Entity Class Diagram

Enum Class Diagram

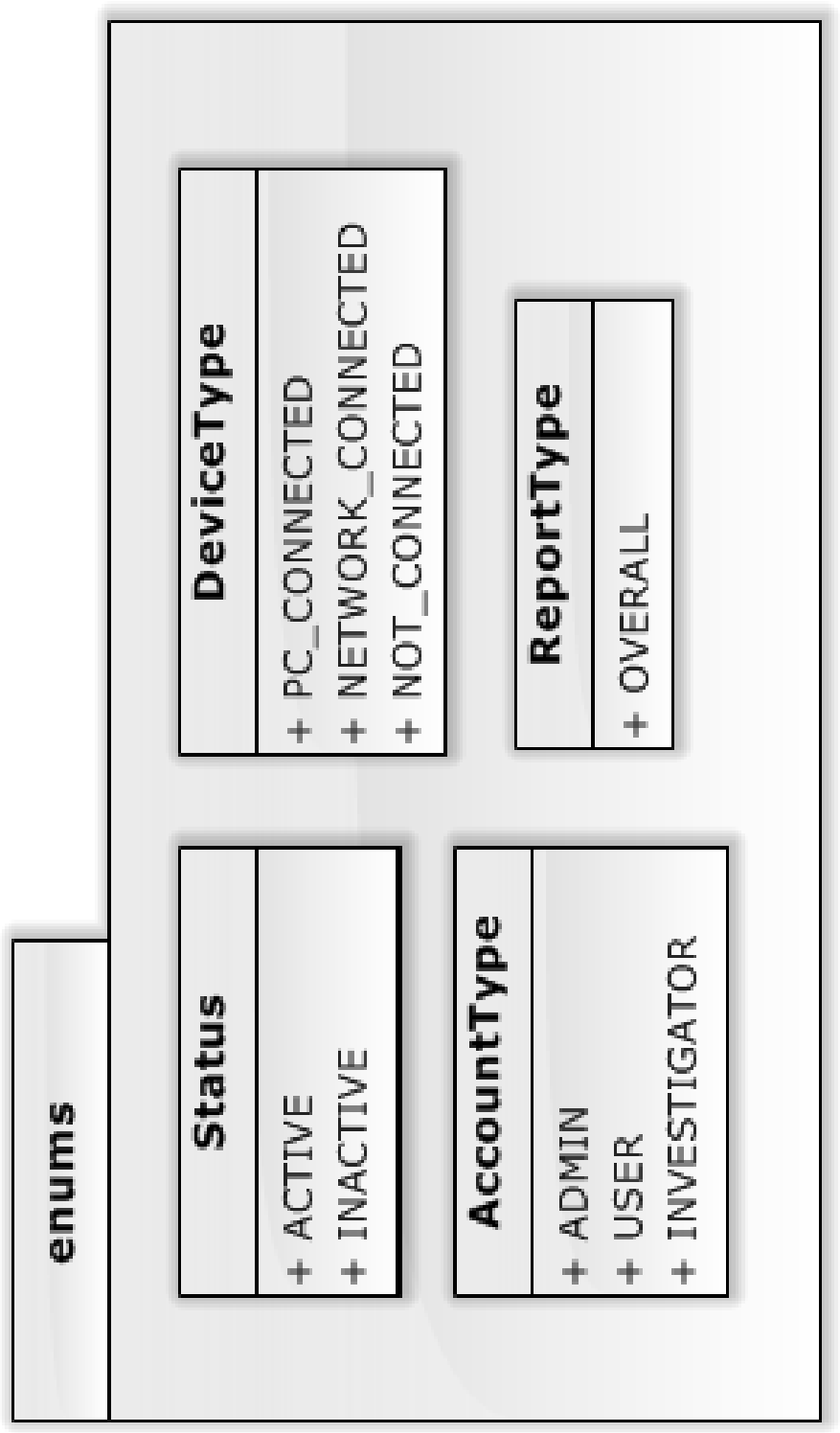

Figure 6 - Enum Class Diagram

Repository Class Diagram

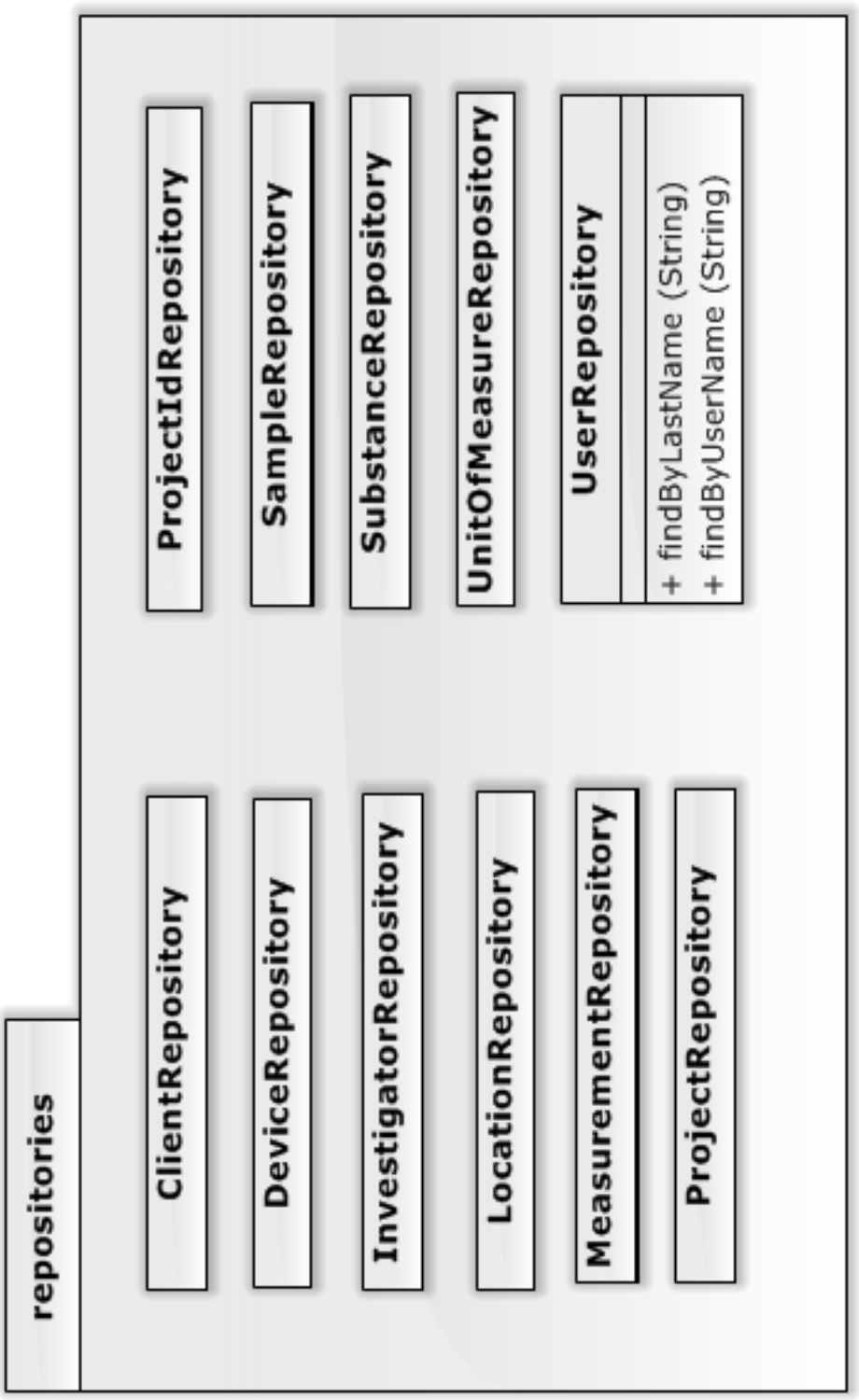

Figure 7 - Repository Class Diagram

Service Class Diagram

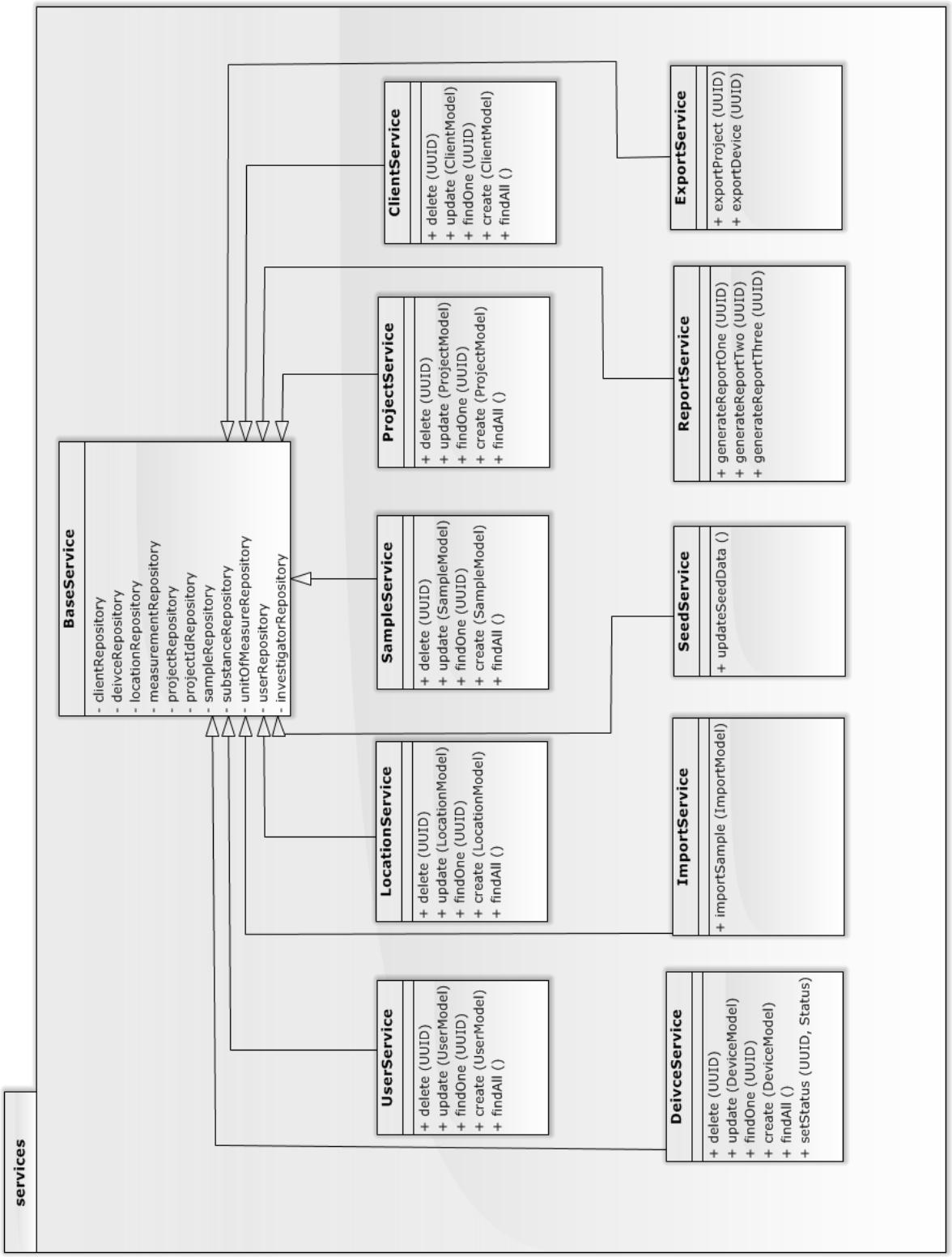

Figure 8 - Service Class Diagram

Task Class Diagram

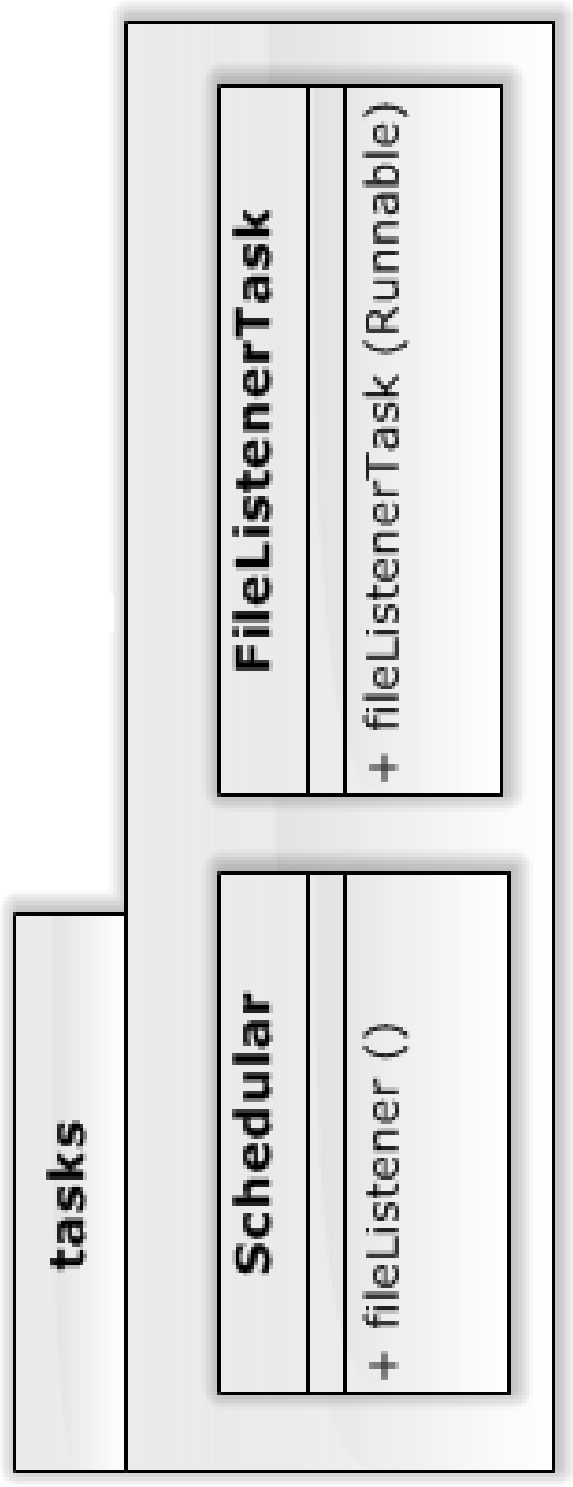

Figure 9 - Task Class Diagram

Interaction Sequence Diagram

Manage Users Sequence Diagram

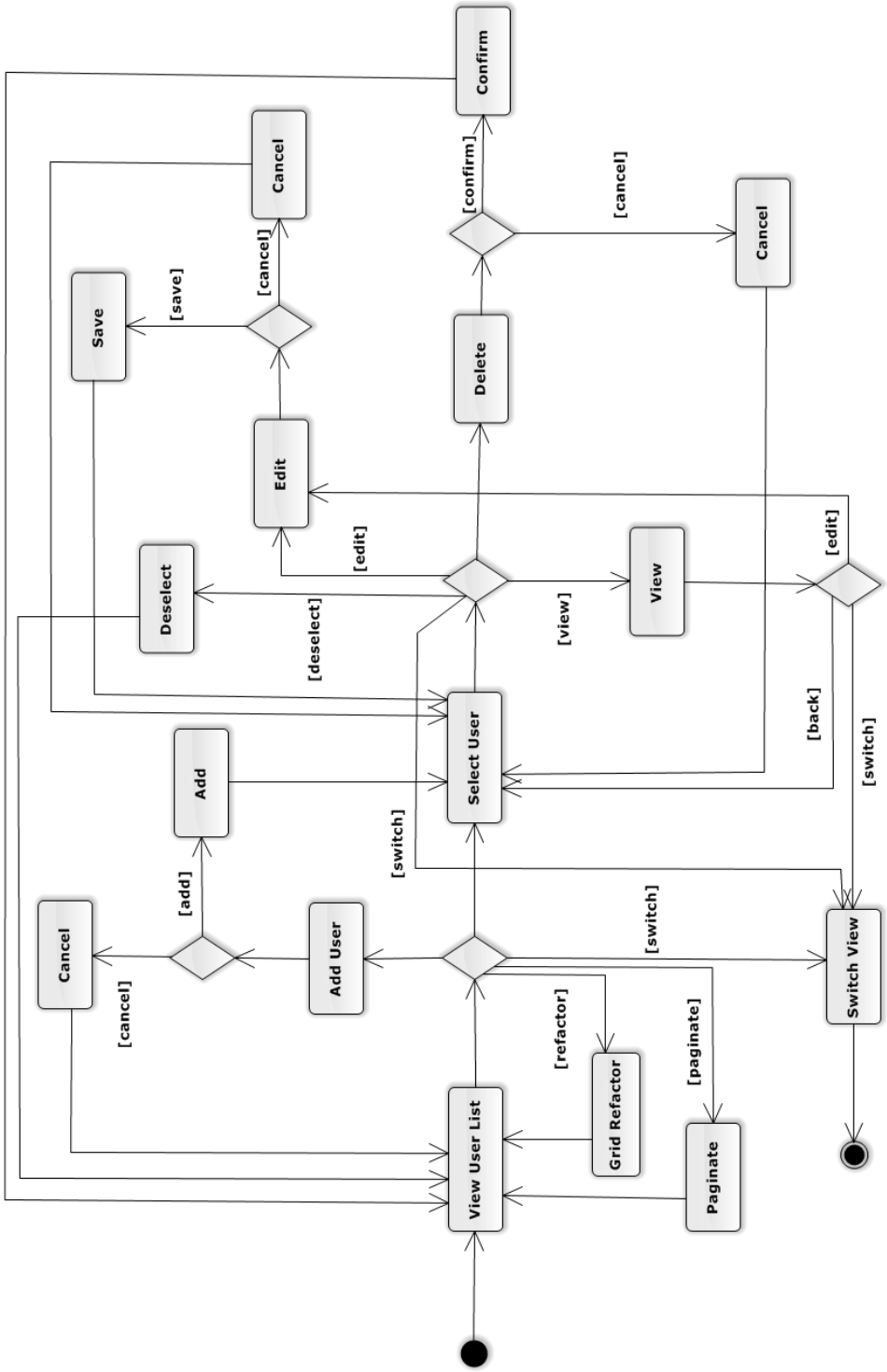

Figure 10 - Manage Users Sequence Diagram

Manage Clients Sequence Diagram

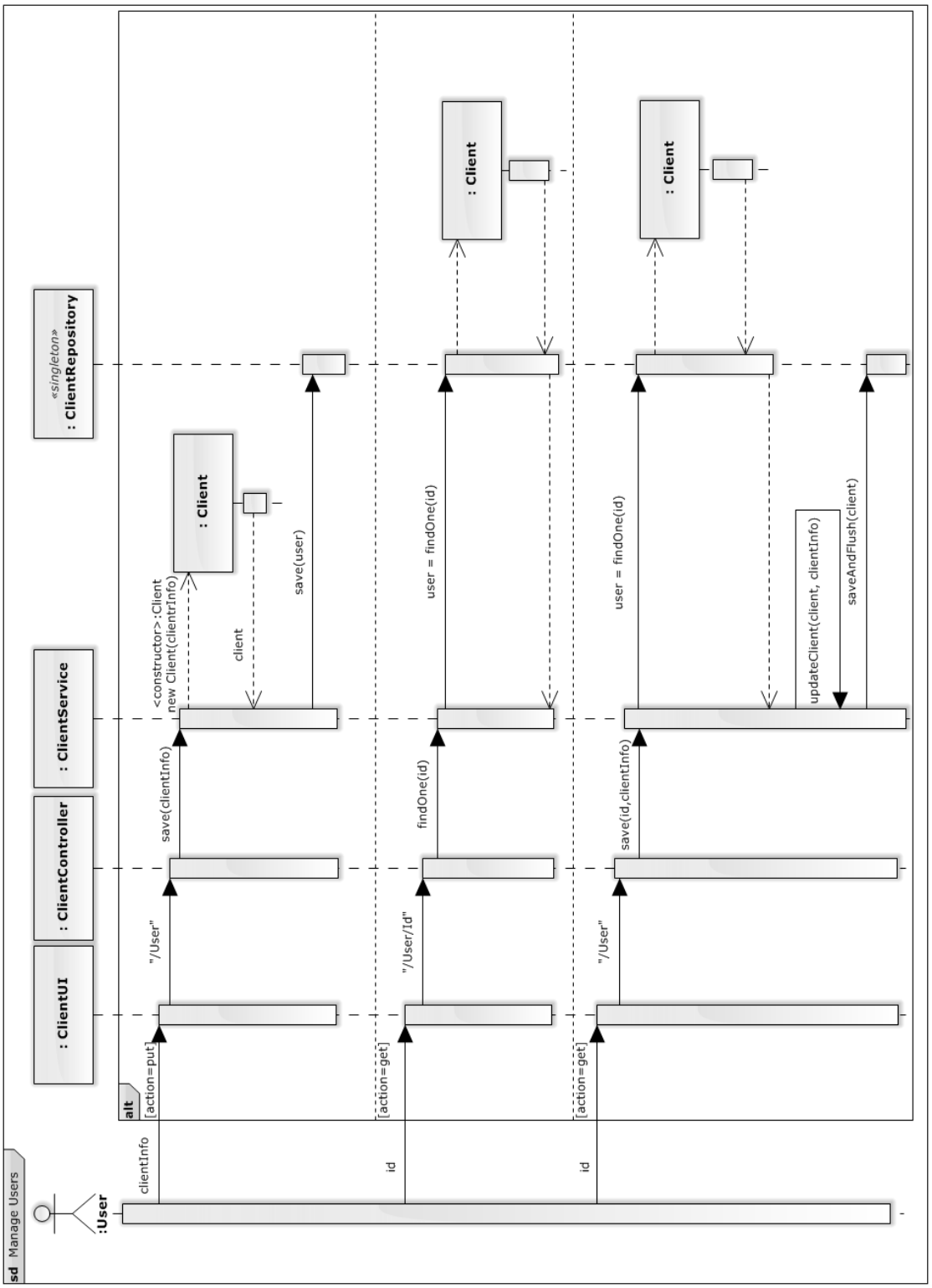

Figure 11 - Manage Clients Sequence Diagram

Login Sequence Diagram

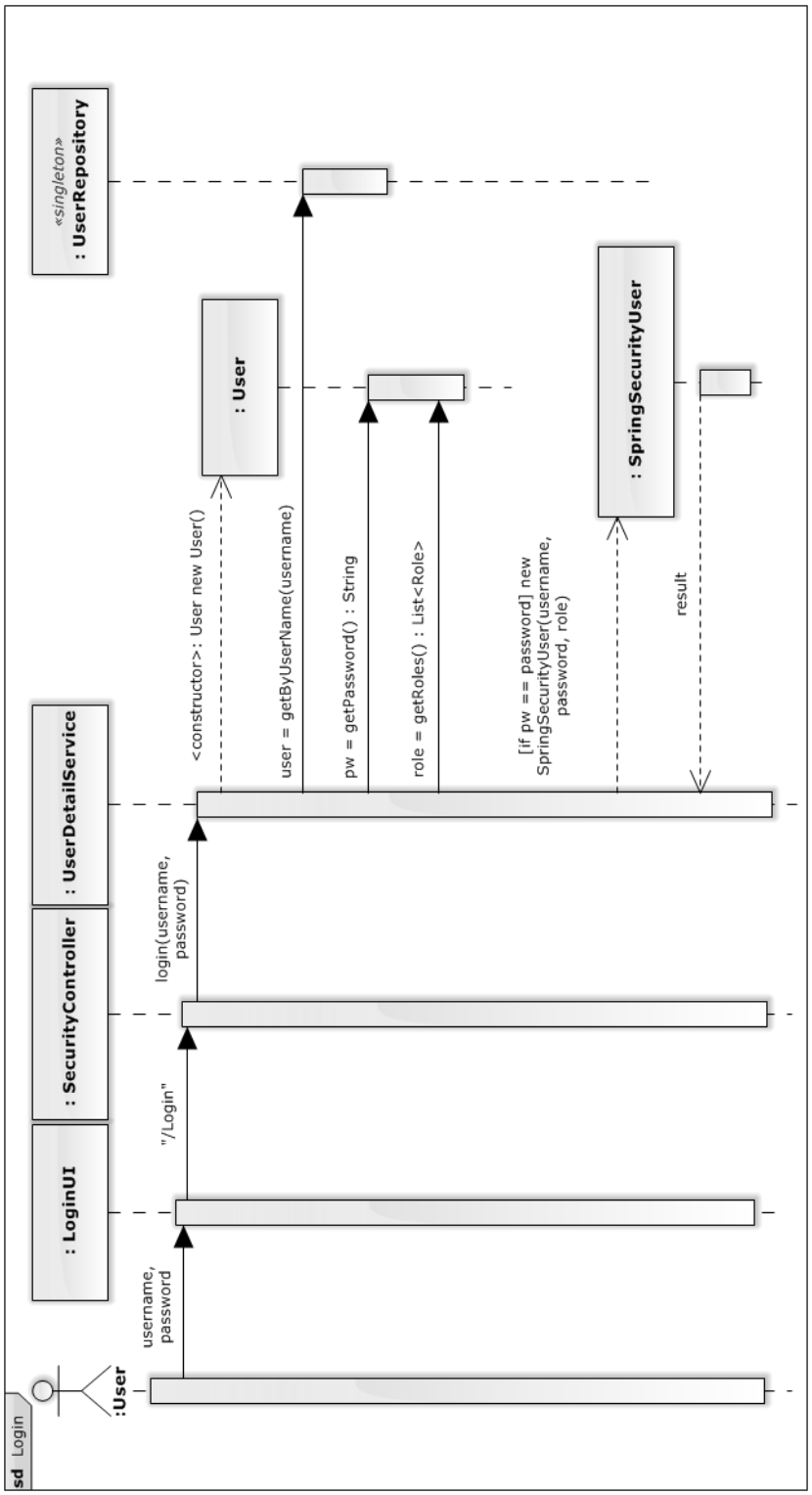

Figure 12 - Login Sequence Diagram

Logout Sequence Diagram

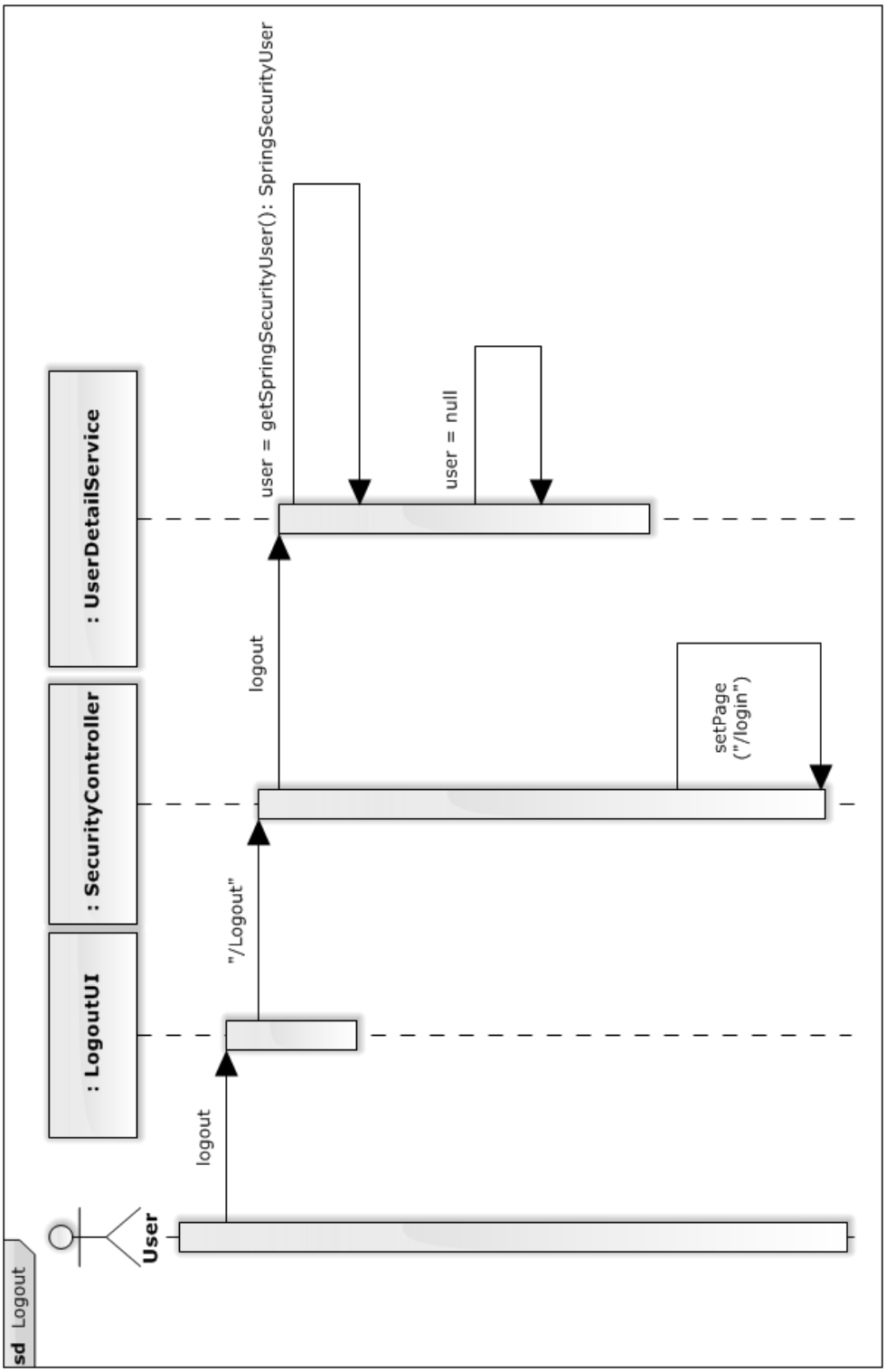

Figure 13 - Logout Sequence Diagram

Permanently Delete Project Sequence Diagram

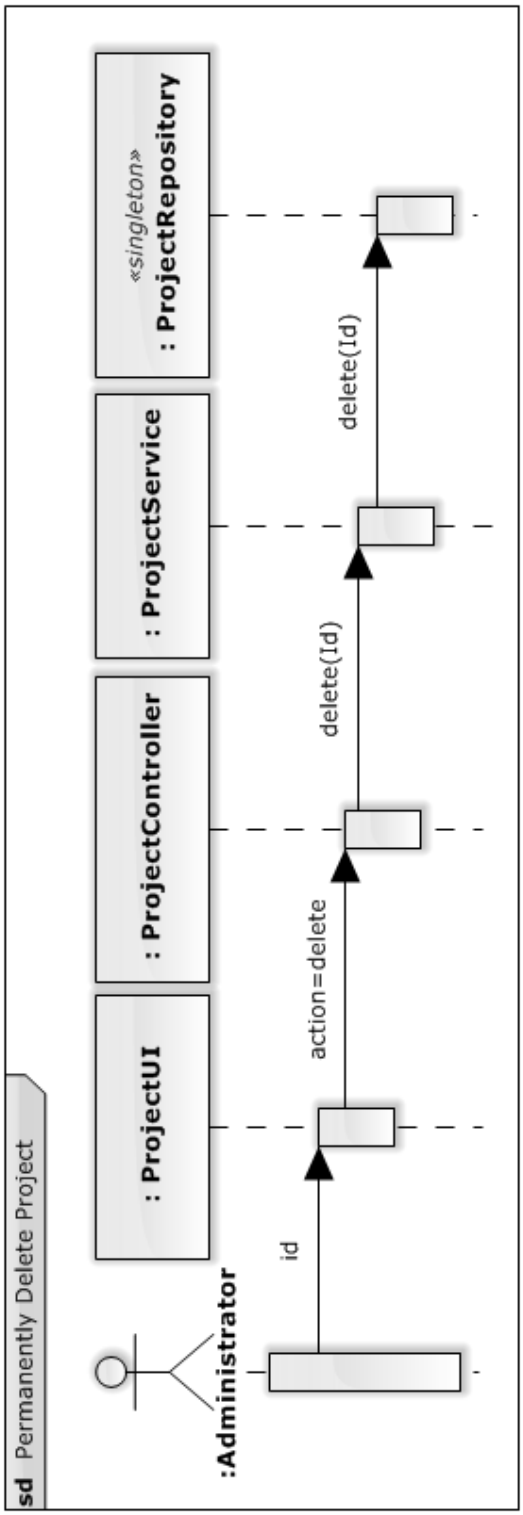

Figure 14 - Permanently Delete Project Sequence Diagram

Import Sample Data Sequence Diagram

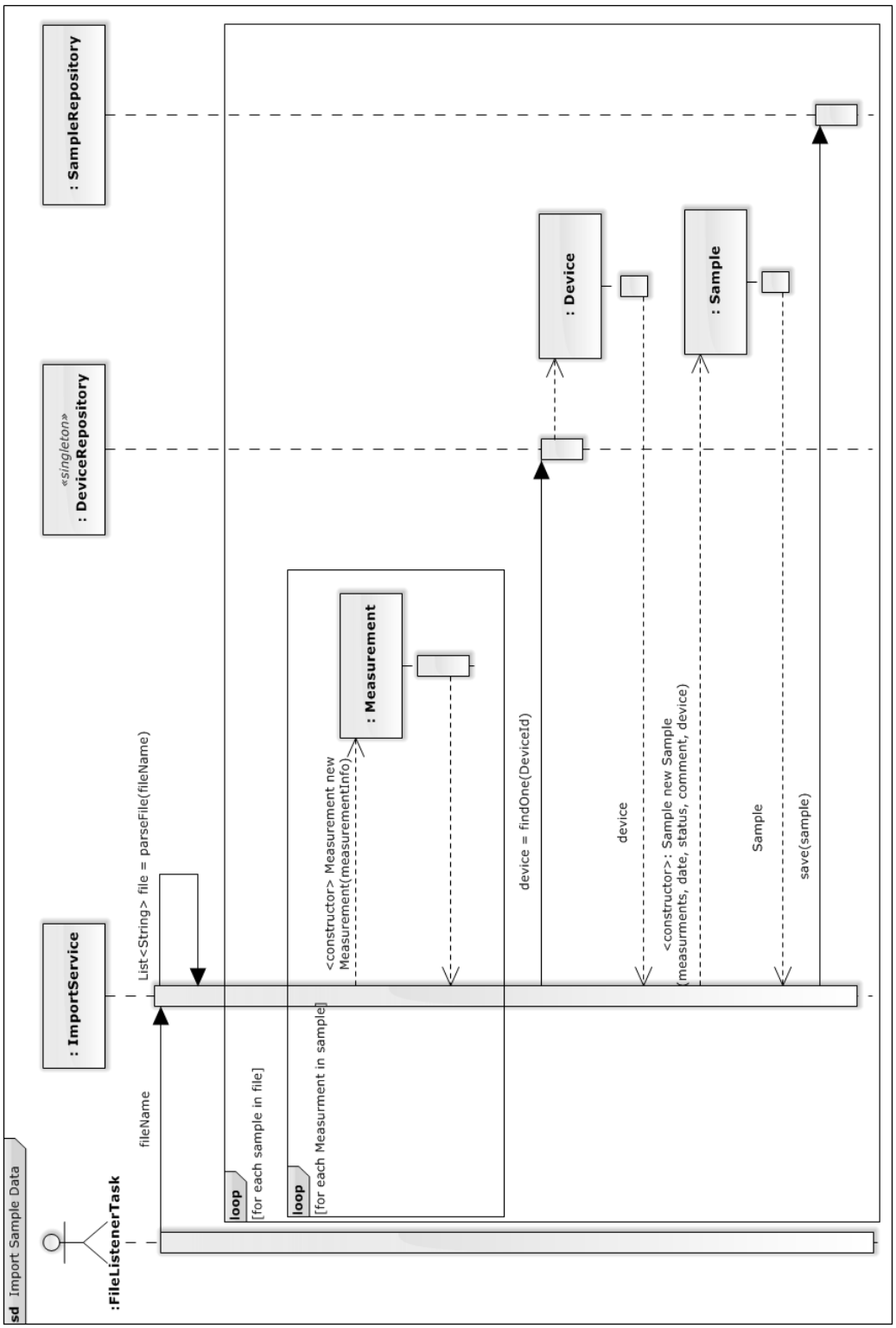

Figure 15 - Import Sample Data Sequence Diagram

Manage Projects Sequence Diagram

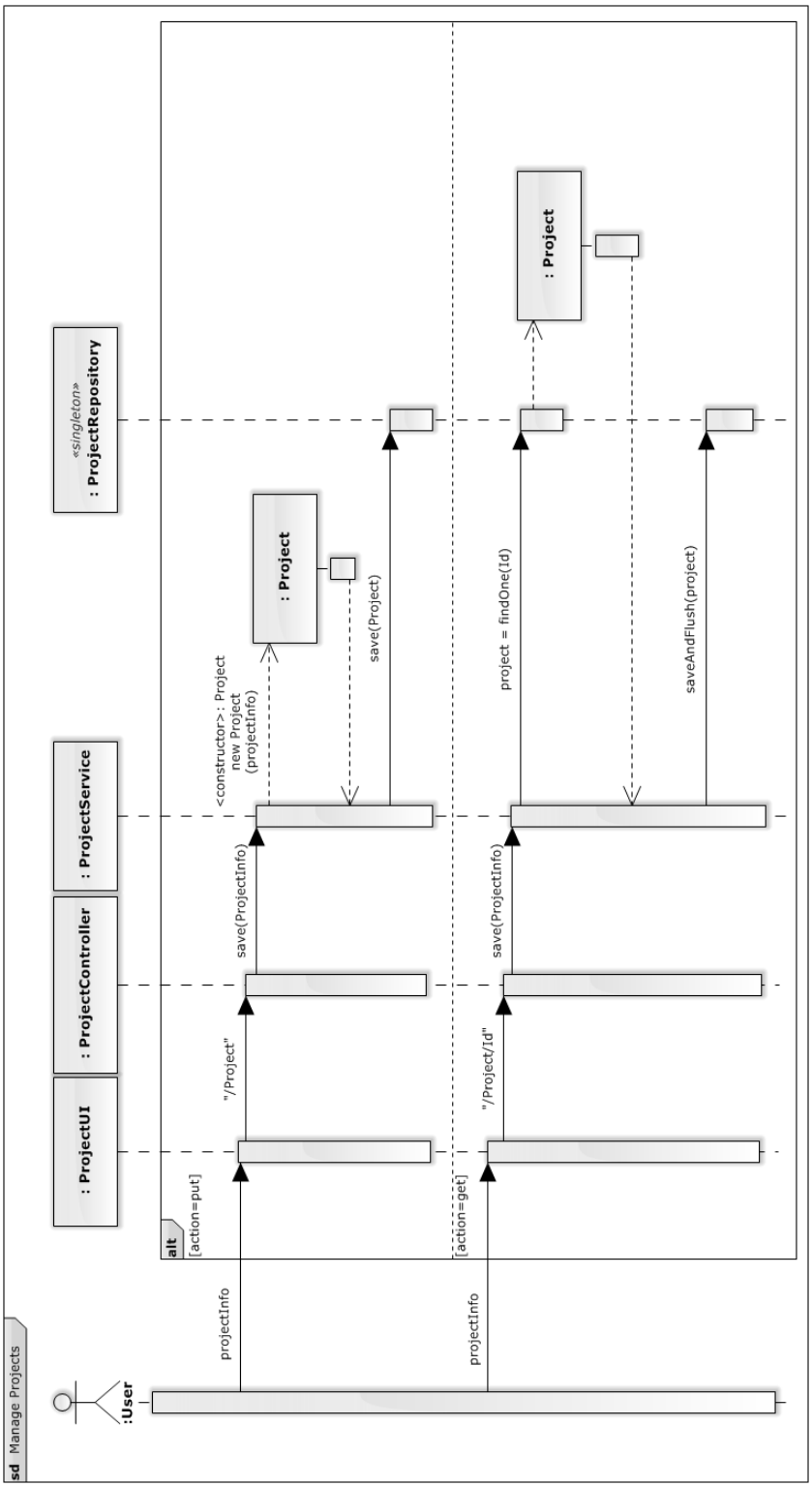

Figure 16 - Manage Project Sample Data Sequence Diagram

View Project Report Sequence Diagram

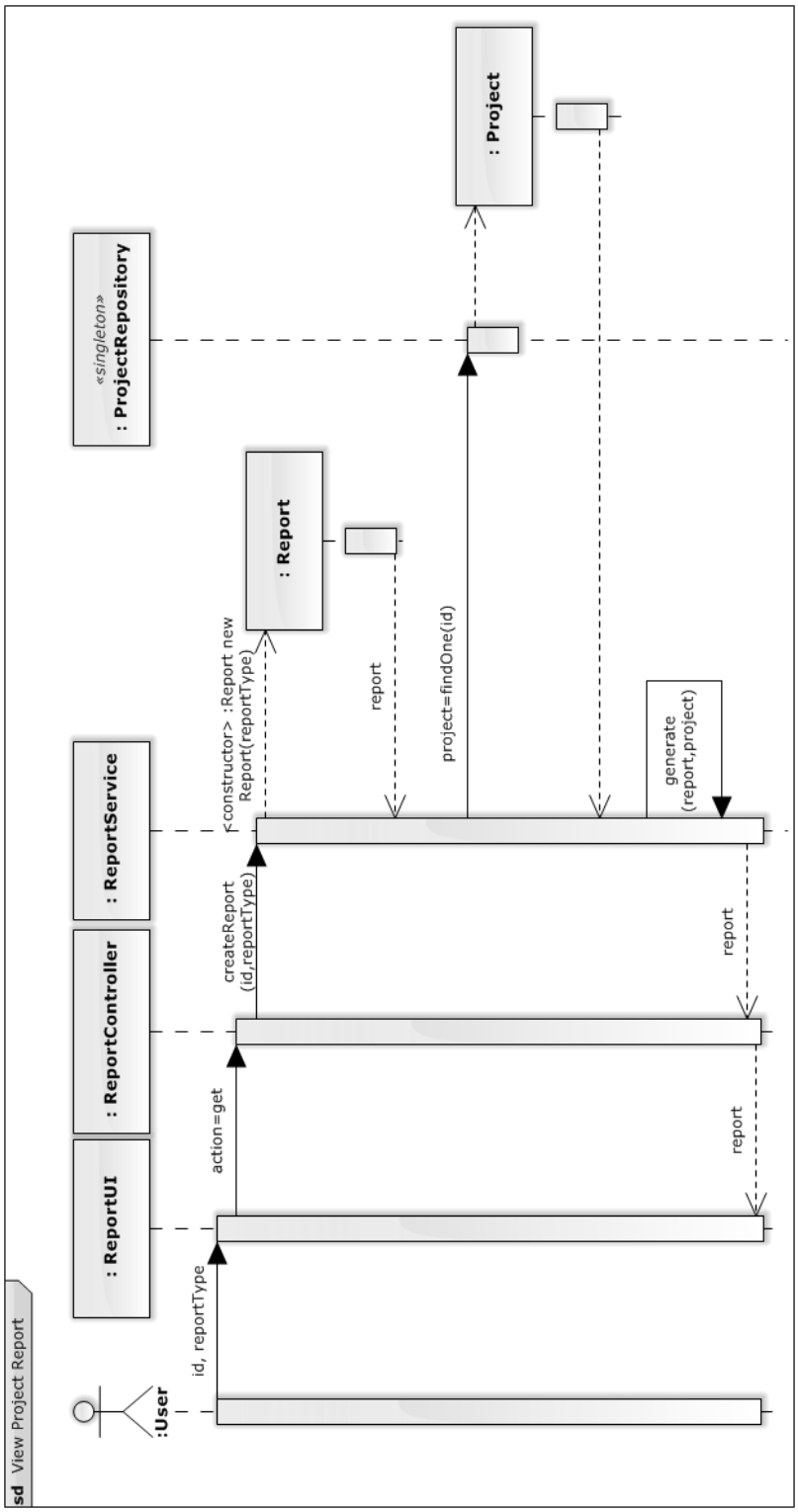

Figure 17 - View Project Report Sequence Diagram

Input Sample Data Sequence Diagram

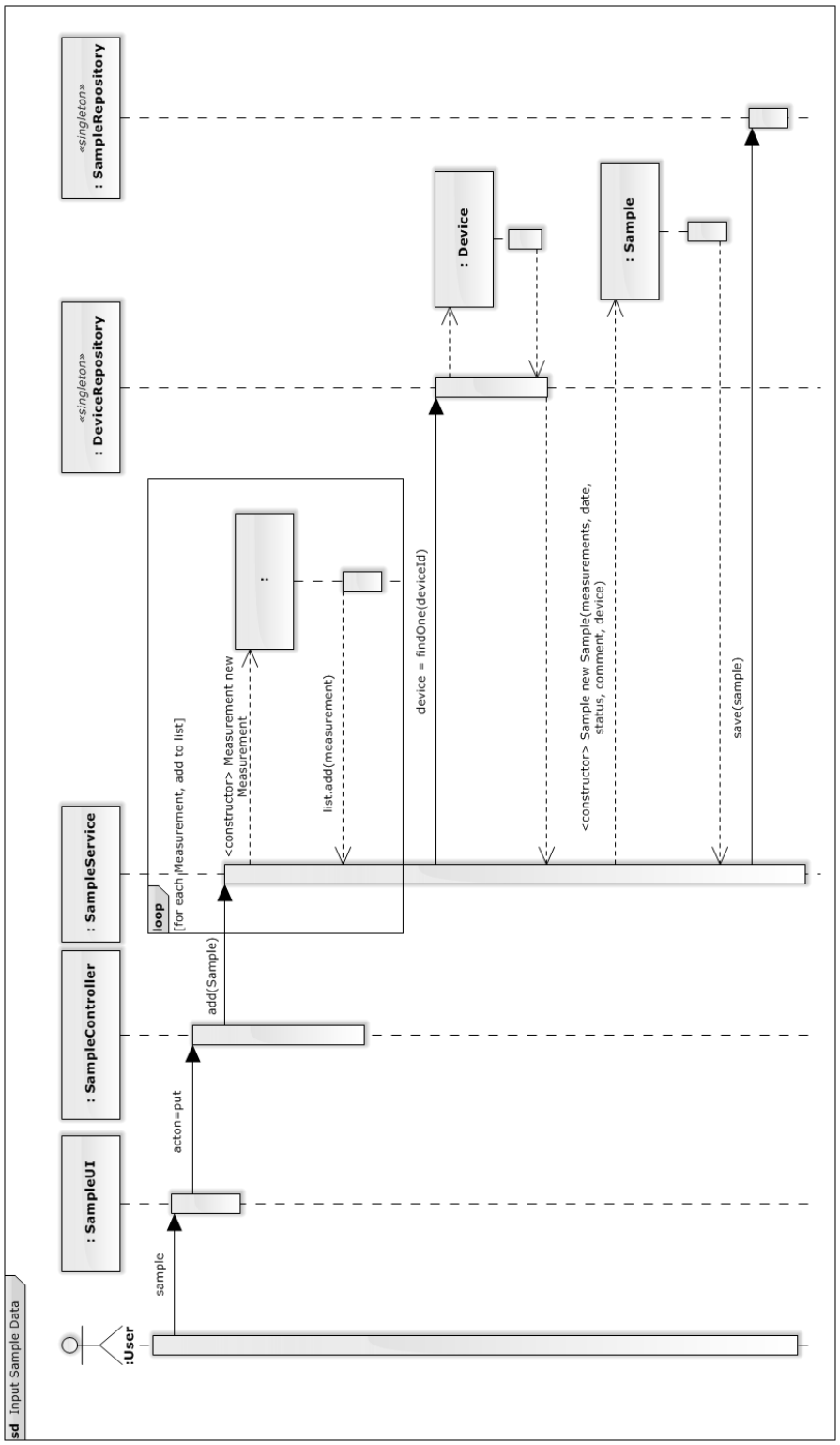

Figure 18 - Input Sample Data Sequence Diagram

Export Data by Project Sequence Diagram

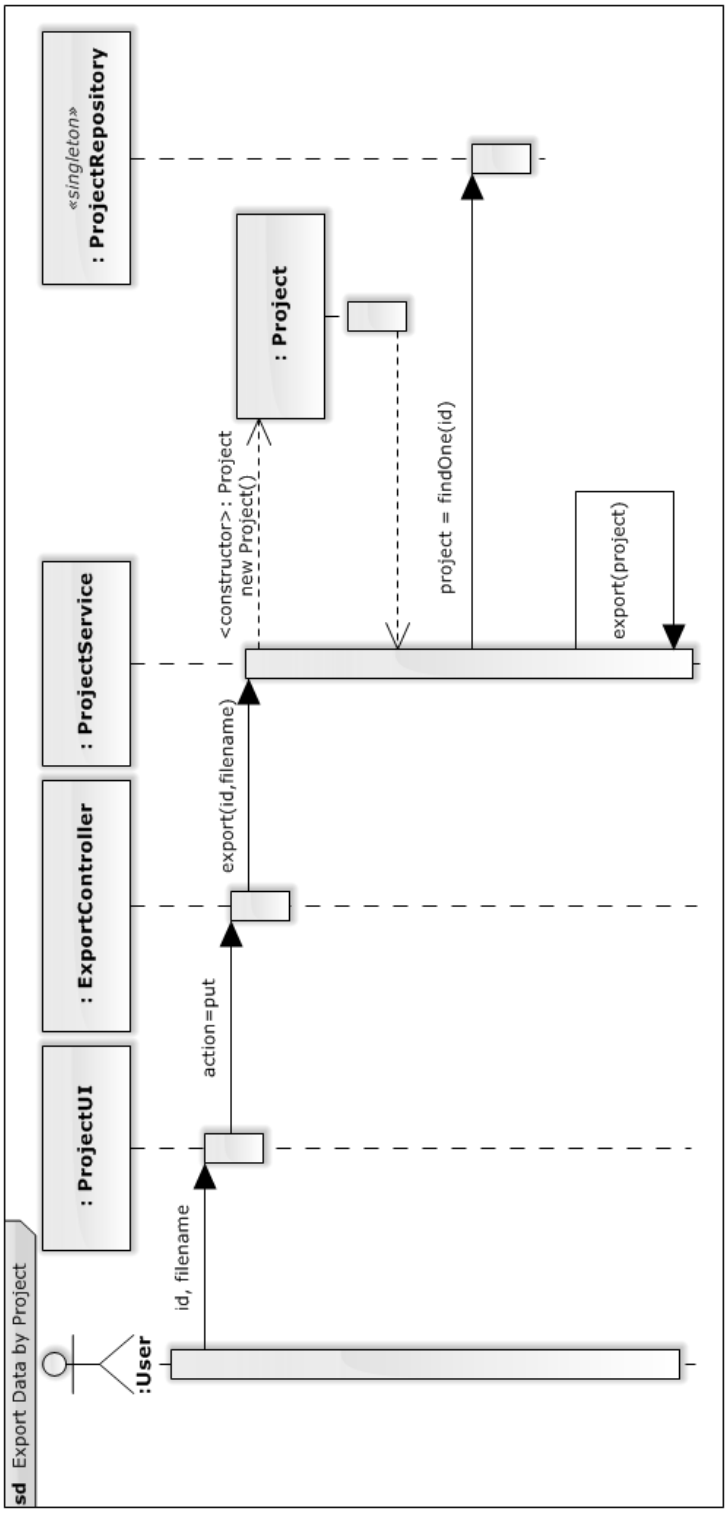

Figure 19 - Export Data by Project Sequence Diagram

Export Data by Device Sequence Diagram

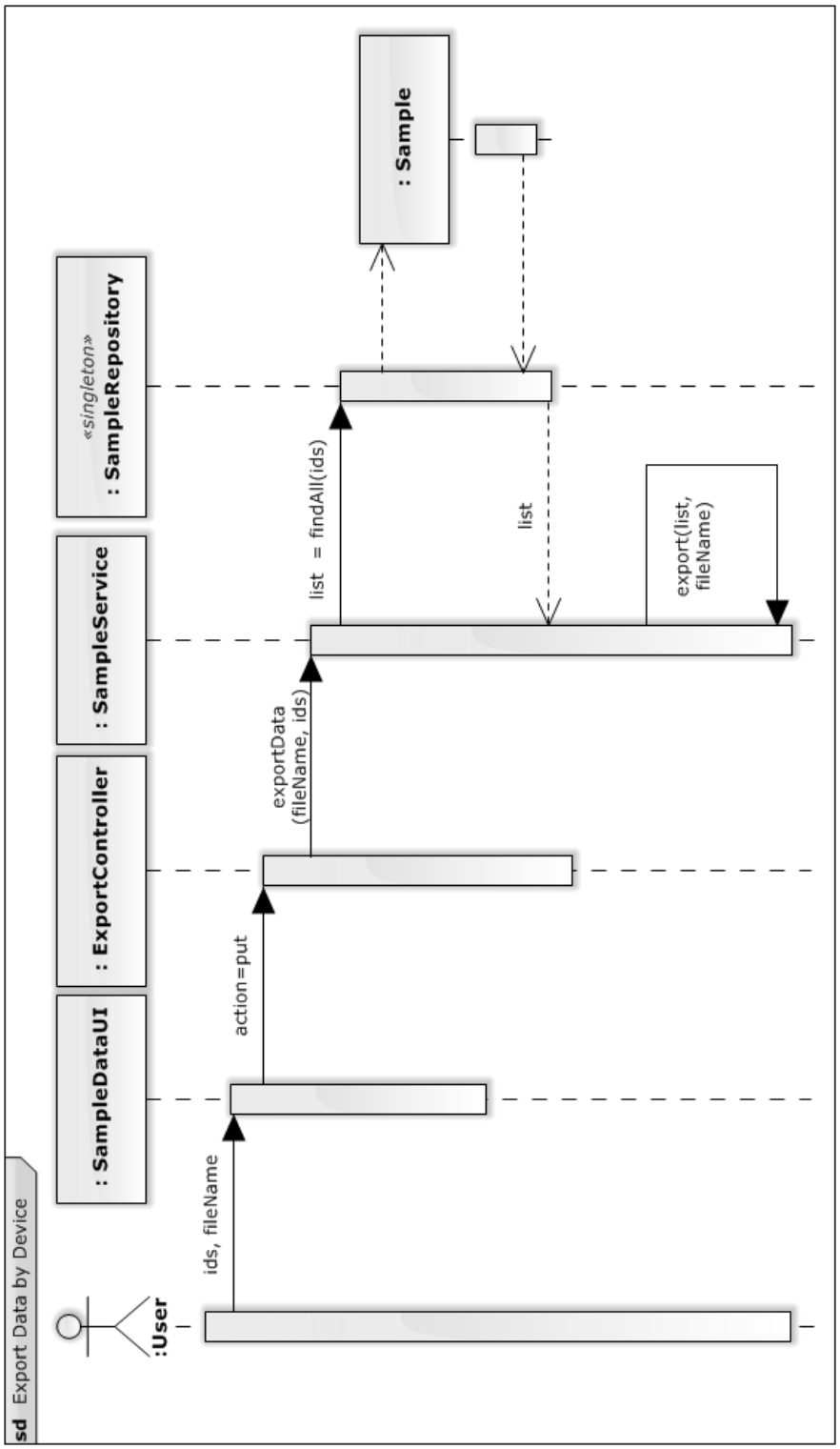

Figure 20 - Export Data by Device Sequence Diagram

Manage Device Sample Data Sequence Diagram

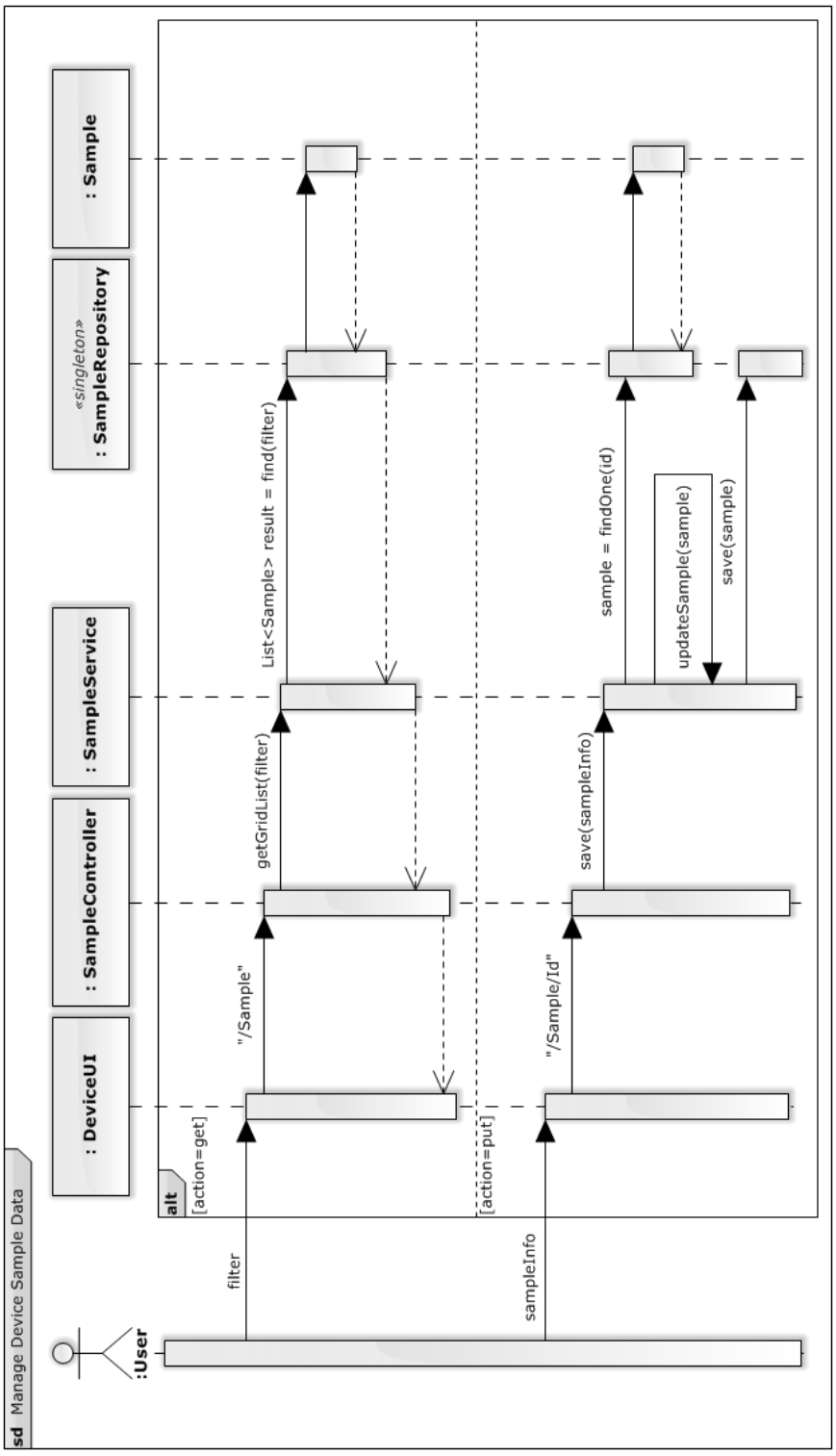

Figure 21 - Manage Device Data Sequence Diagram

Manage Devices Sequence Diagram

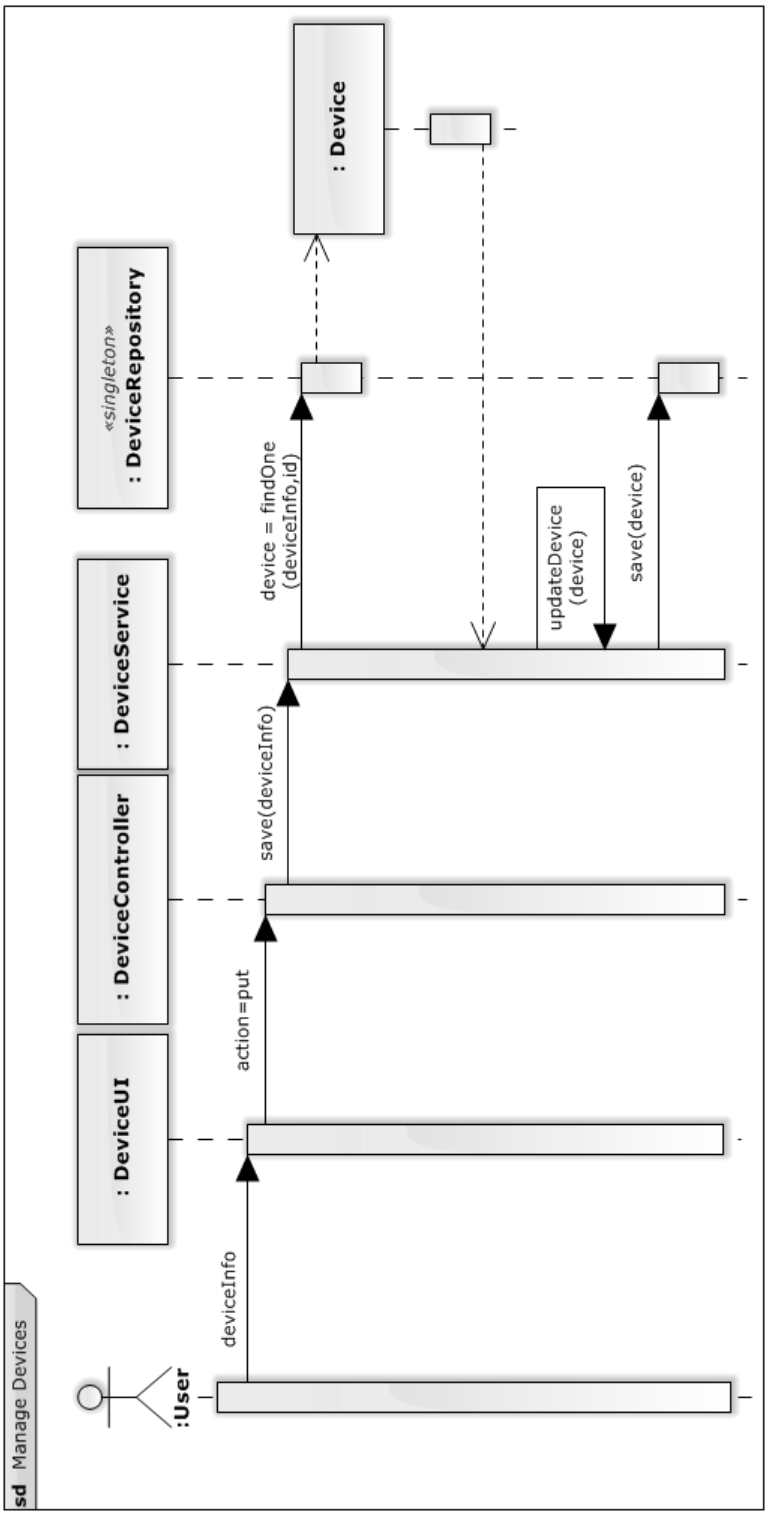

Figure 22 – Manage Devices Sequence Diagram

Import Project Sequence Diagram

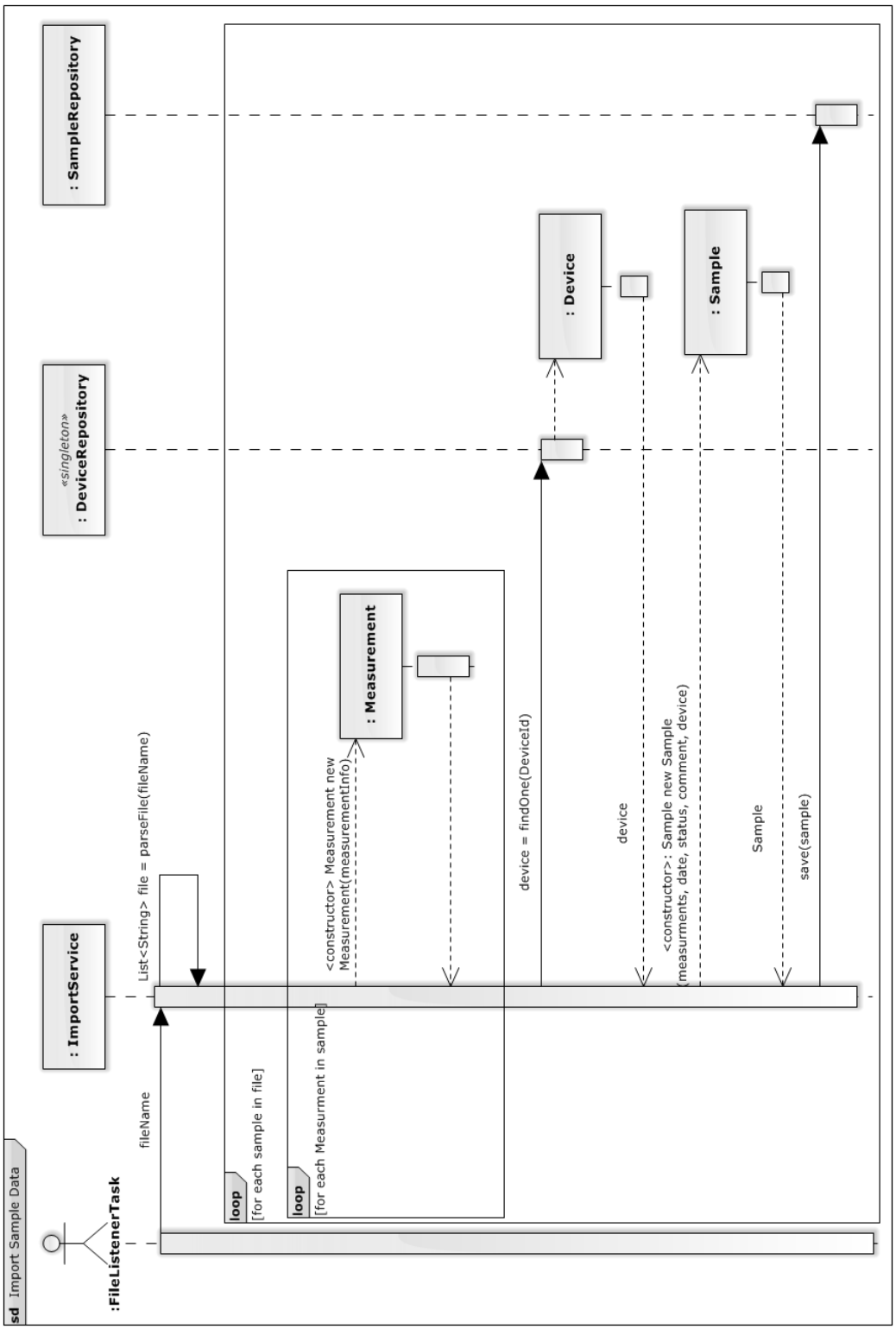

Figure 23 - Import Project Sequence Diagram

Manage Project Sample Data Sequence Diagram

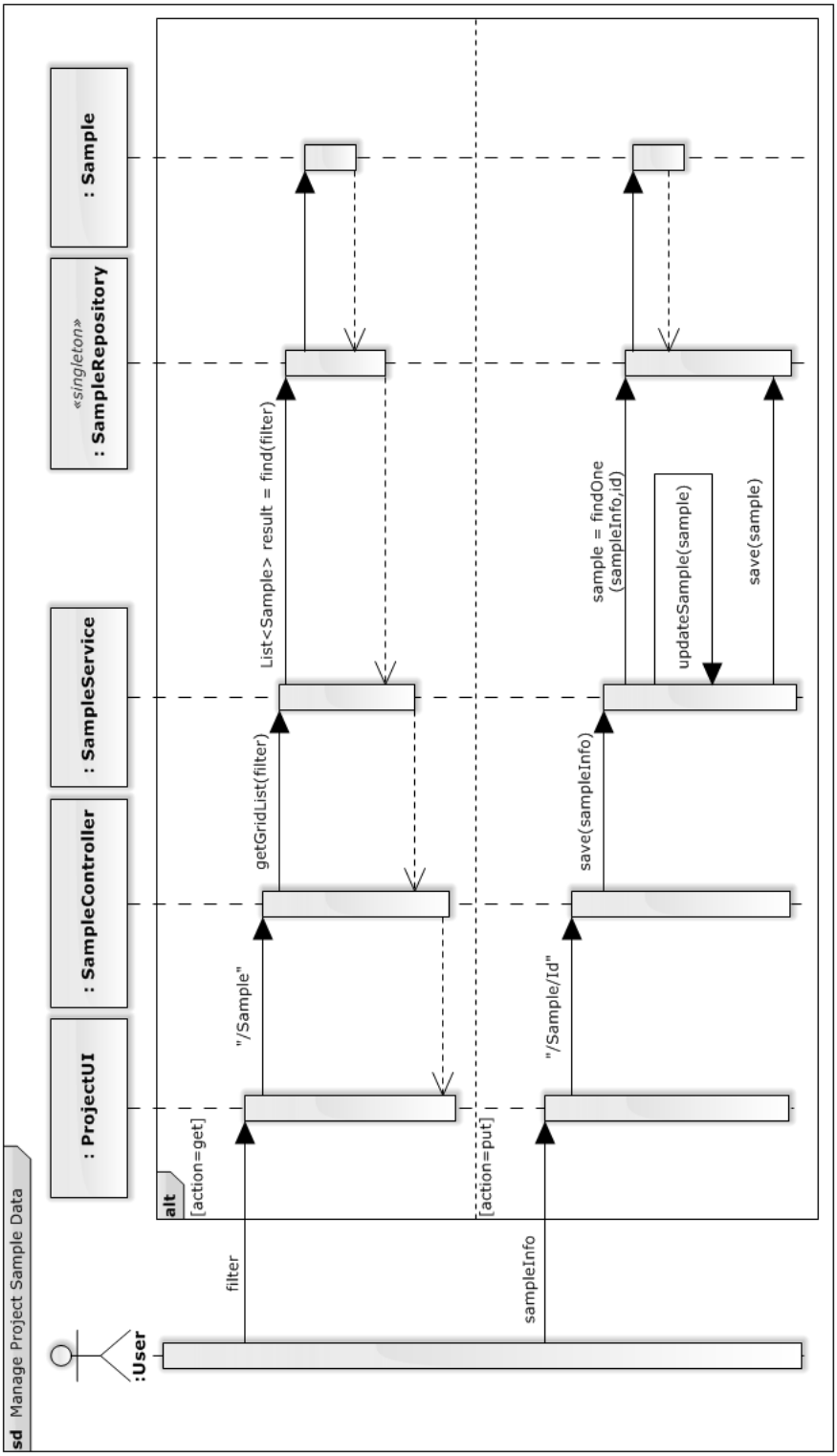

Figure 24 - Manage Project Sample Data Sequence Diagram

State Machine Diagram

Page Navigation State Machine

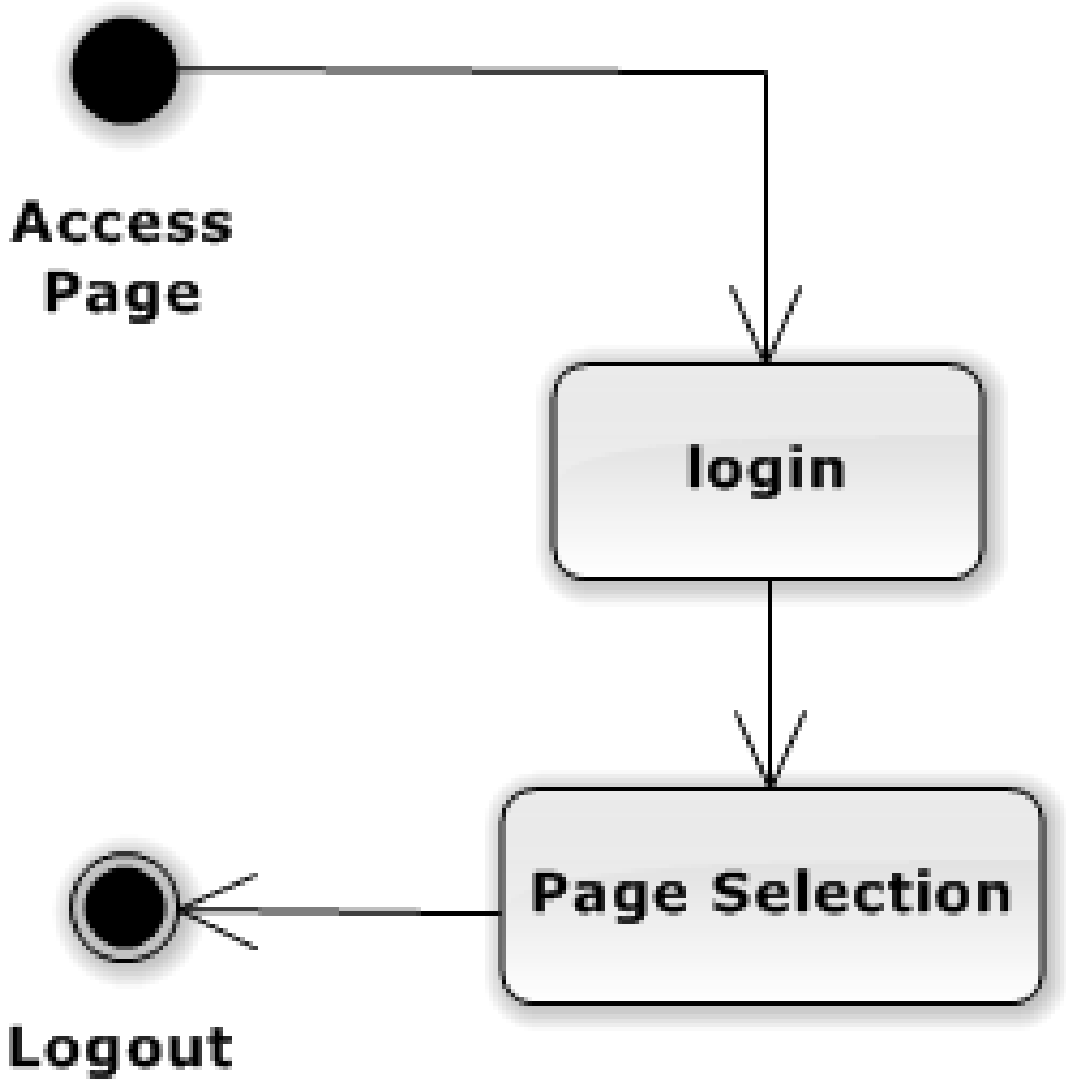

Figure 25 - Page Navigation State Machine

Project State Machine

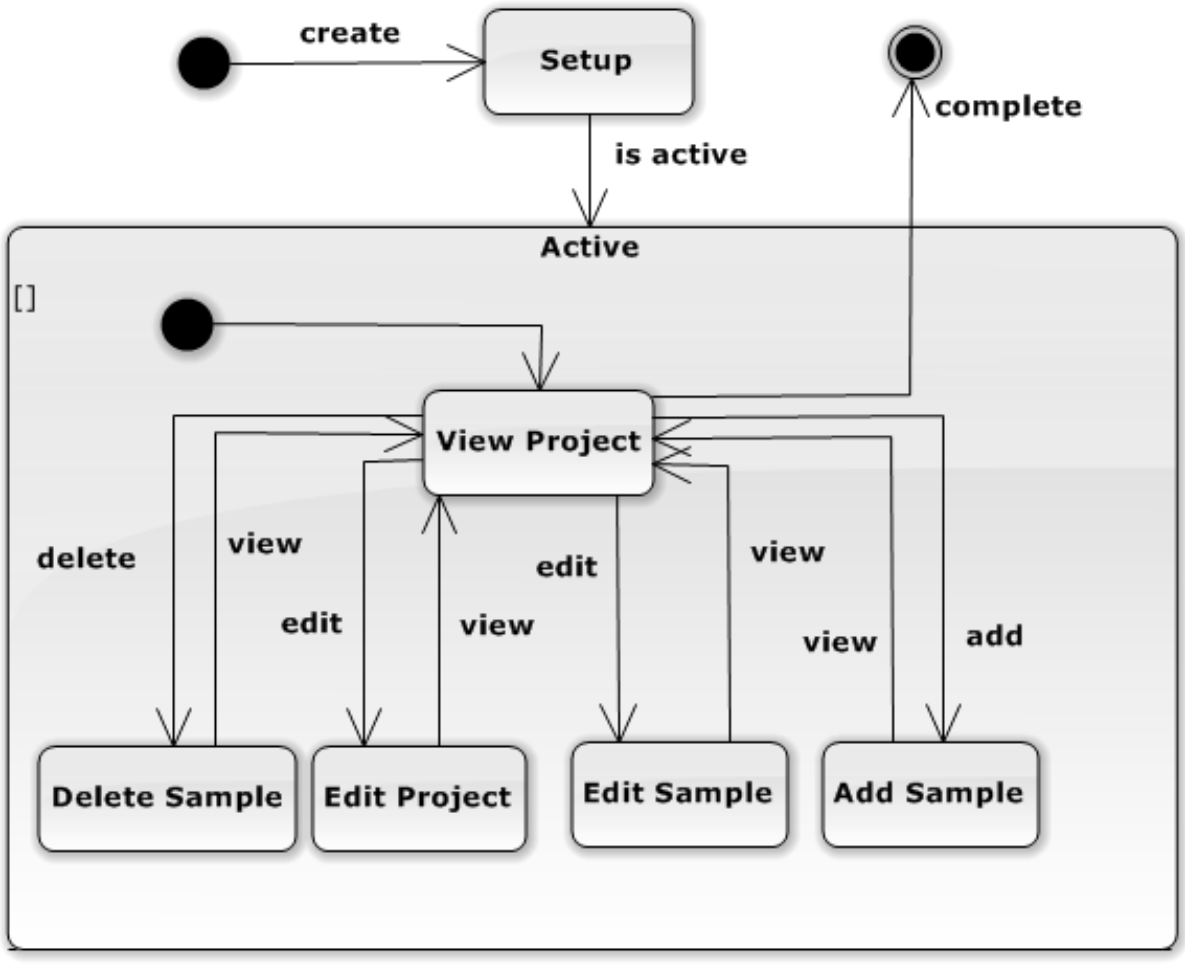

Figure 26 - Project State Machine

Sample State Machine

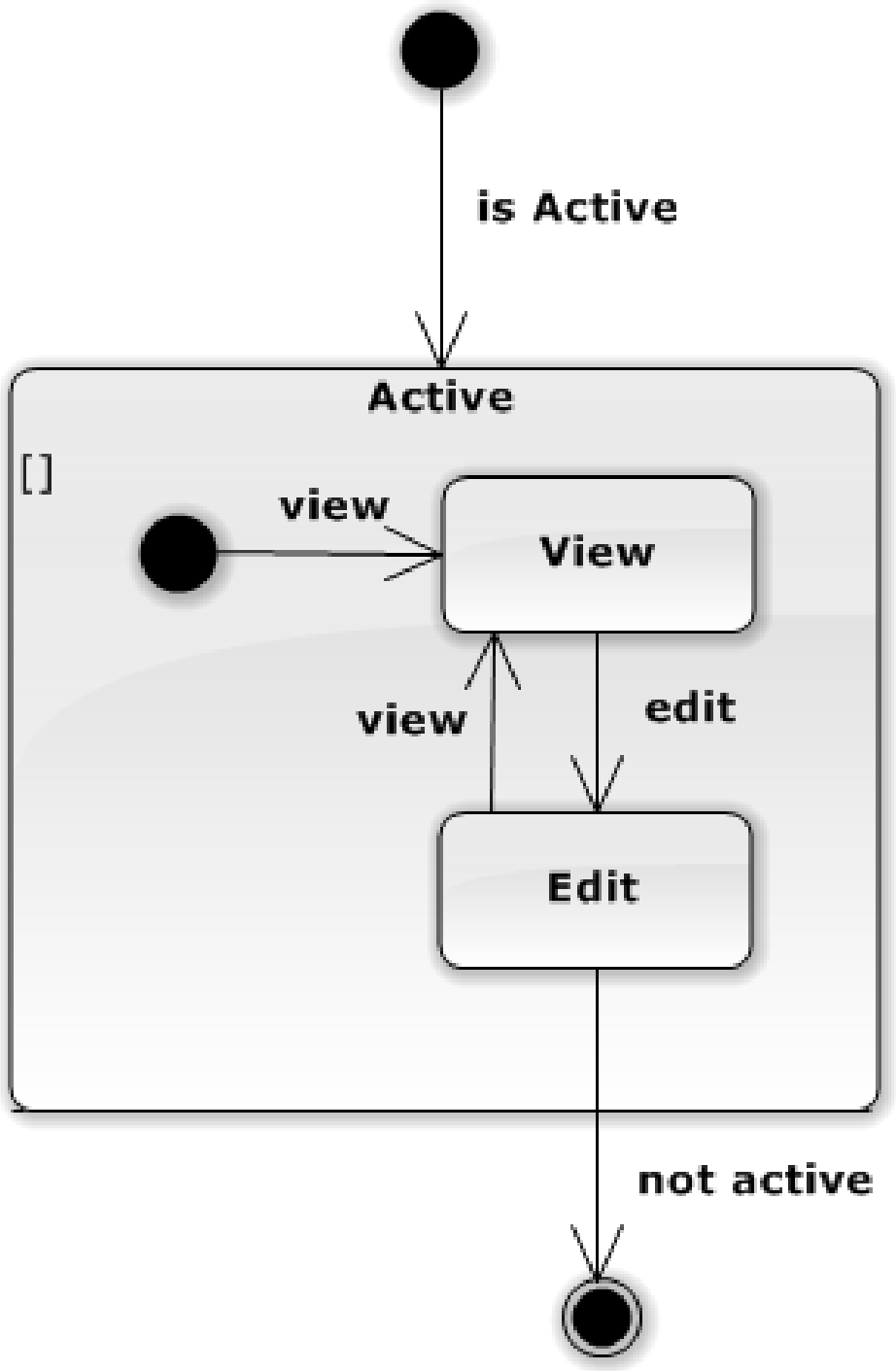

Figure 27 - Sample State Machine

Device State Machine

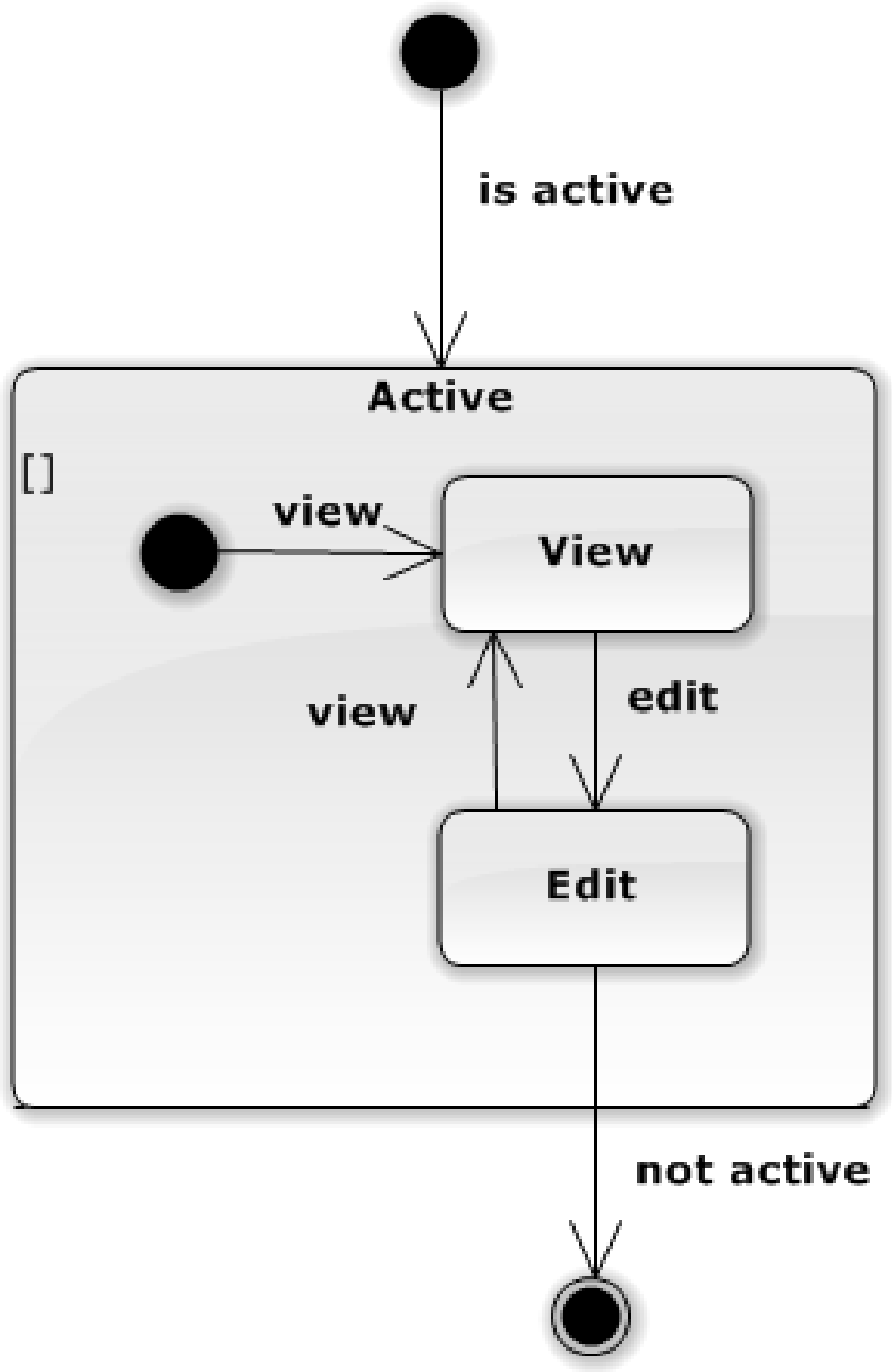

Figure 28 - Device State Machine

User State Machine

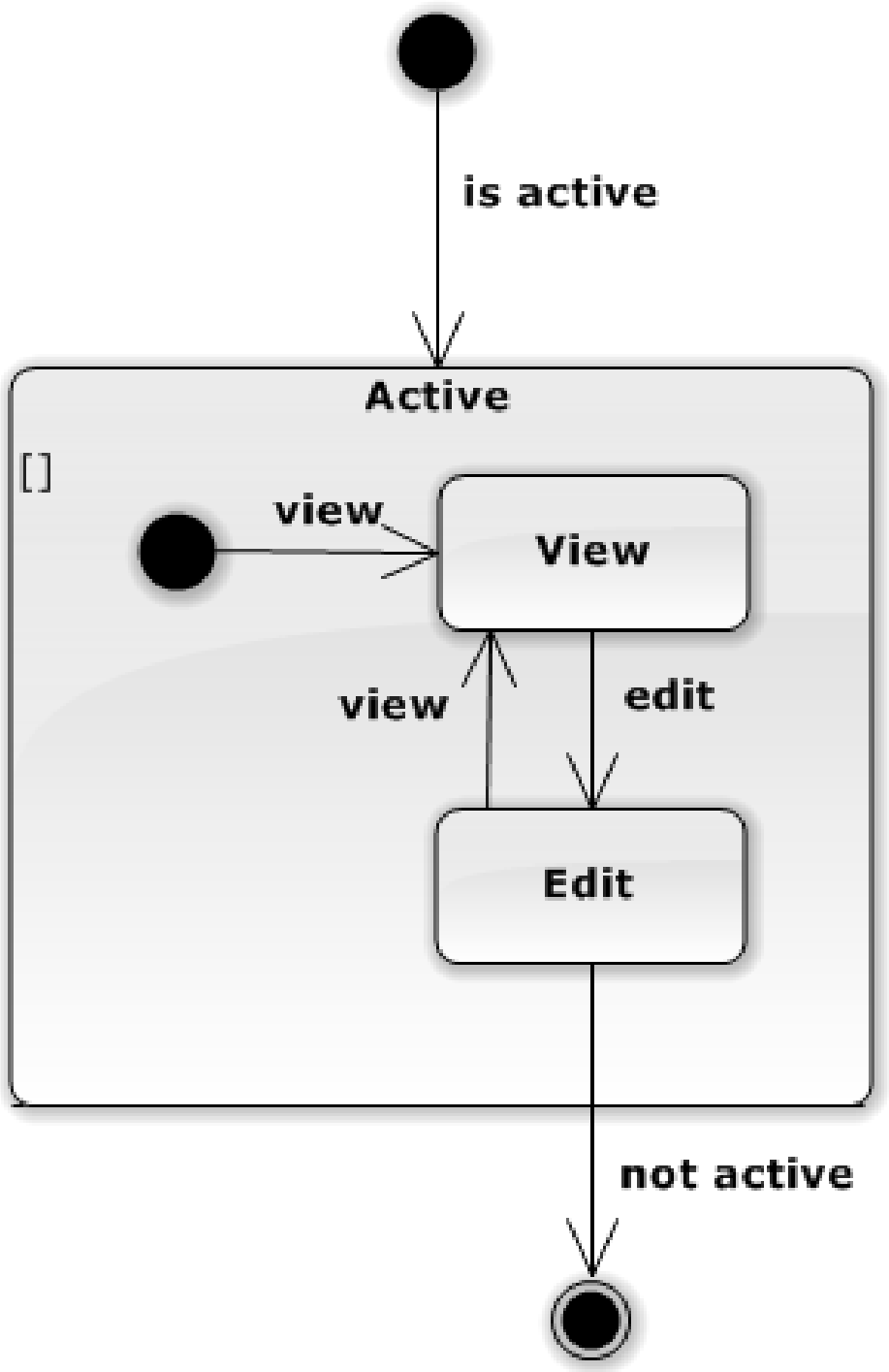

Figure 29 - User State Machine

Activity Diagram

Login Activity Diagram

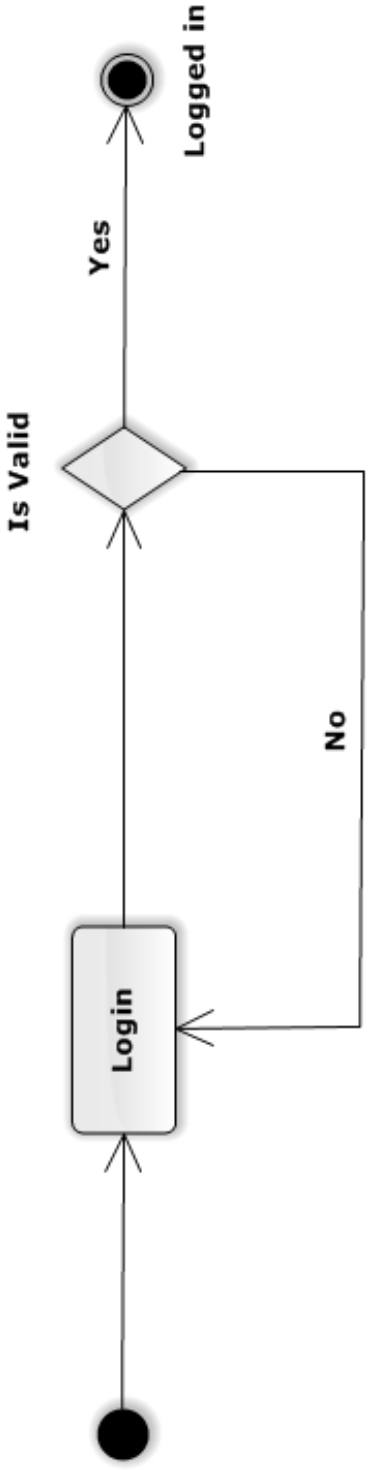

Figure 30 - Login Activity Diagram

Delete Project Activity Diagram

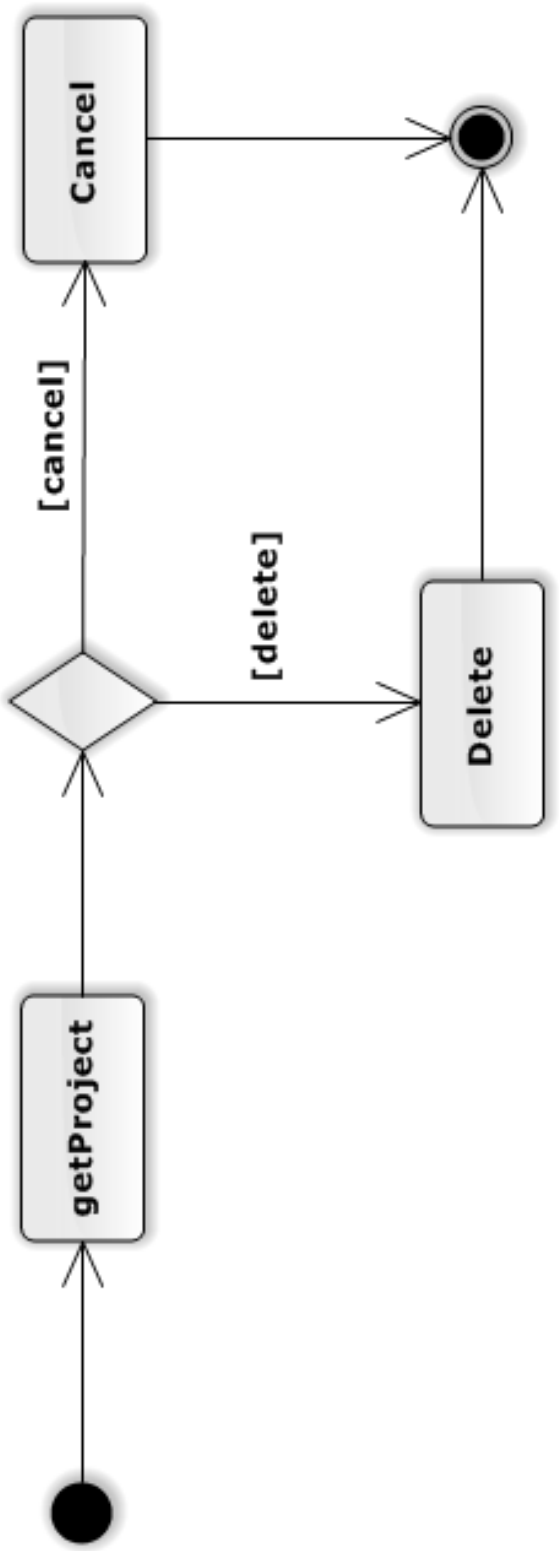

Figure 31 - Delete Project Activity Diagram

Import Project Activity Diagram

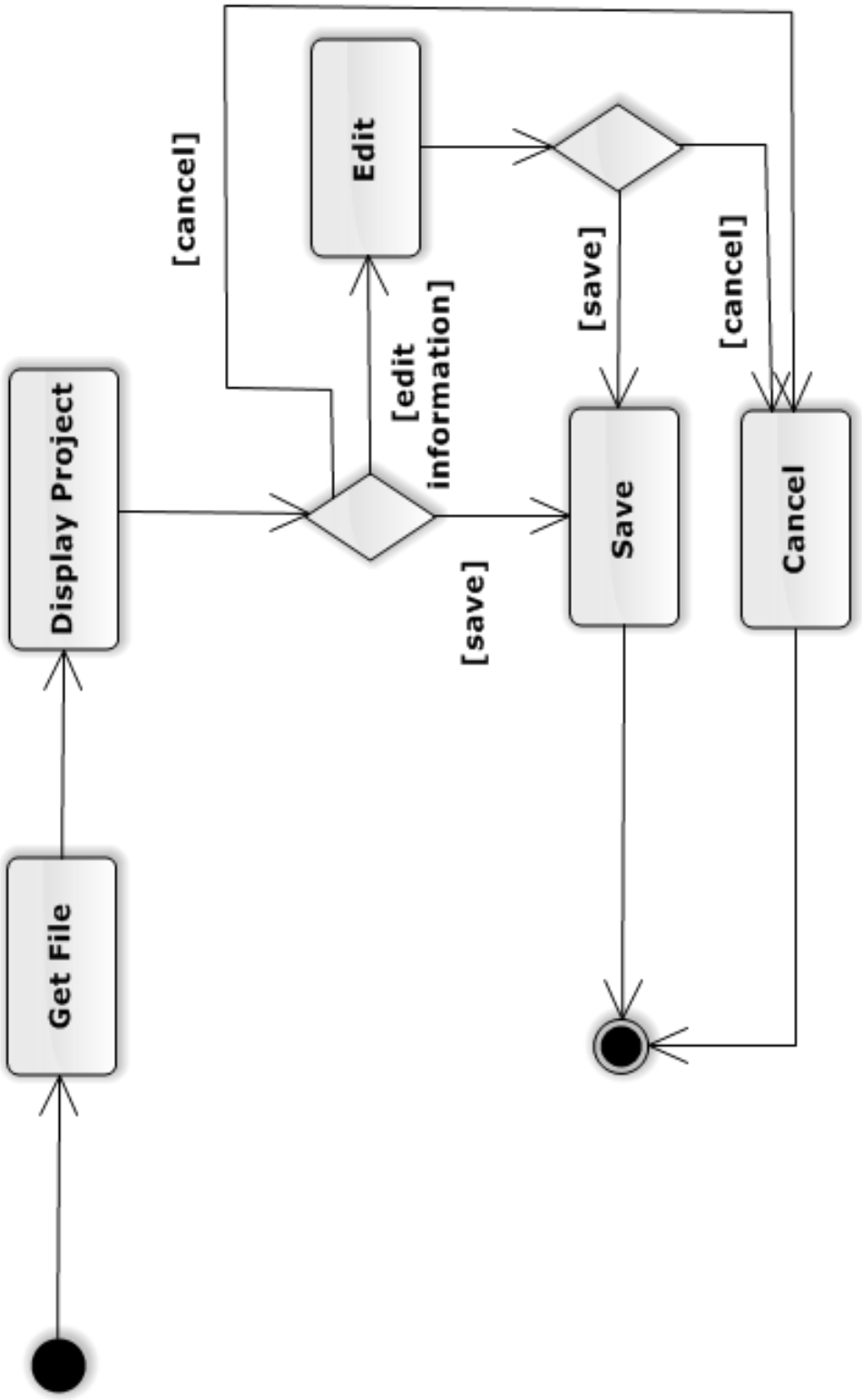

Figure 32 - Import Project Activity Diagram

View Report Activity Diagram

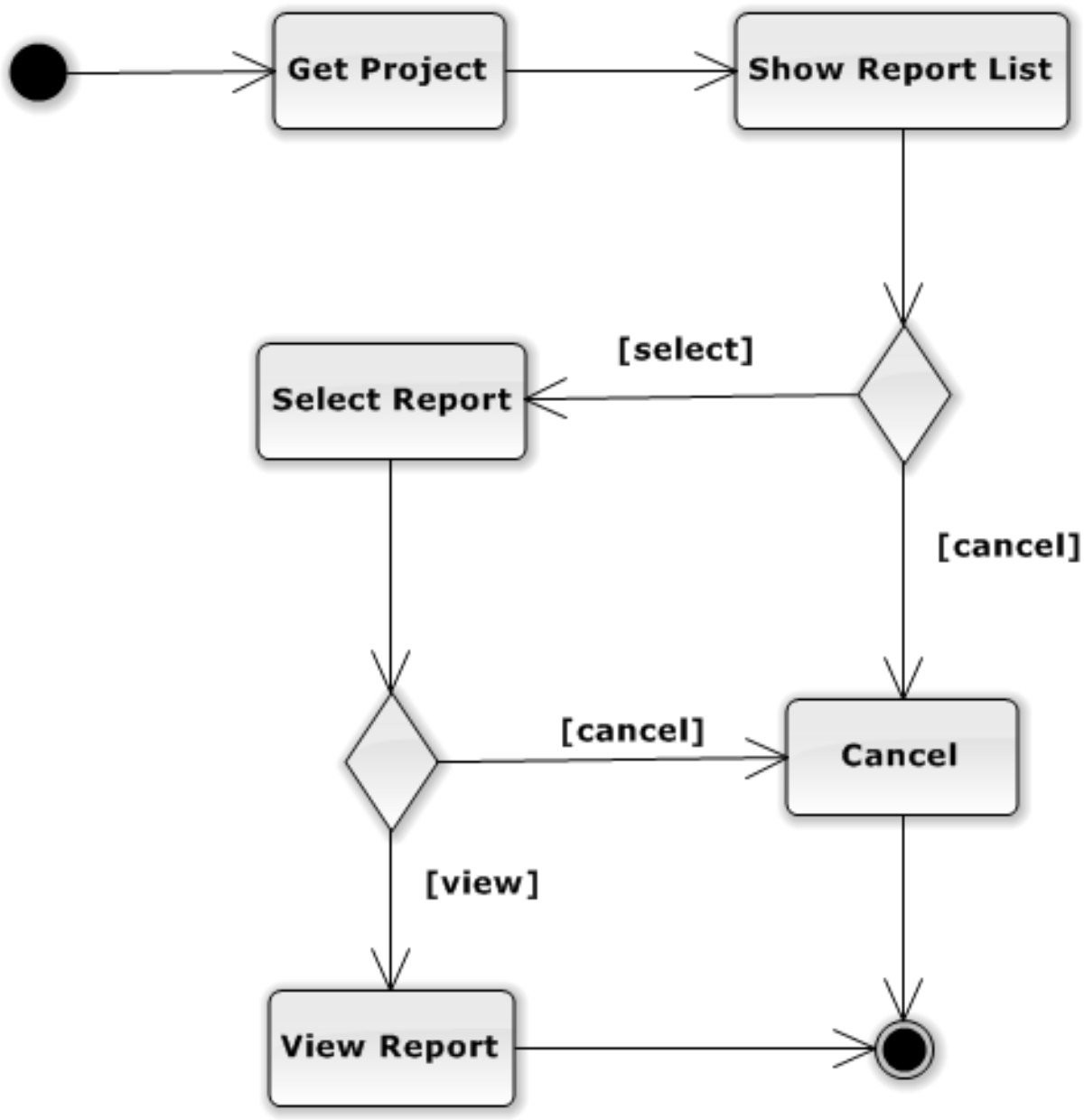

Figure 33 - View Report Activity Diagram

Export Data Activity Diagram

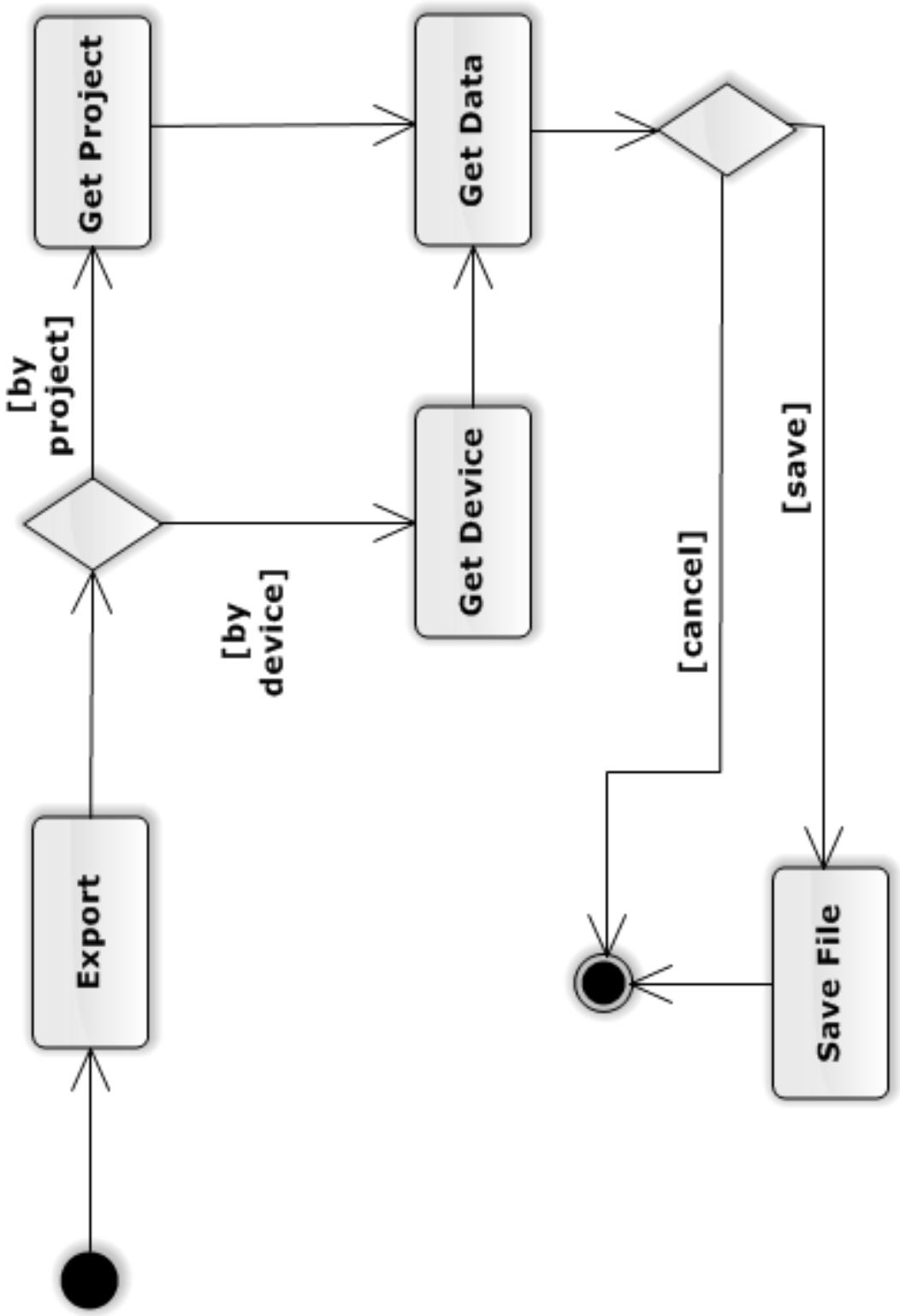

Figure 34 - Export Data Activity Diagram

Input Data Activity Diagram

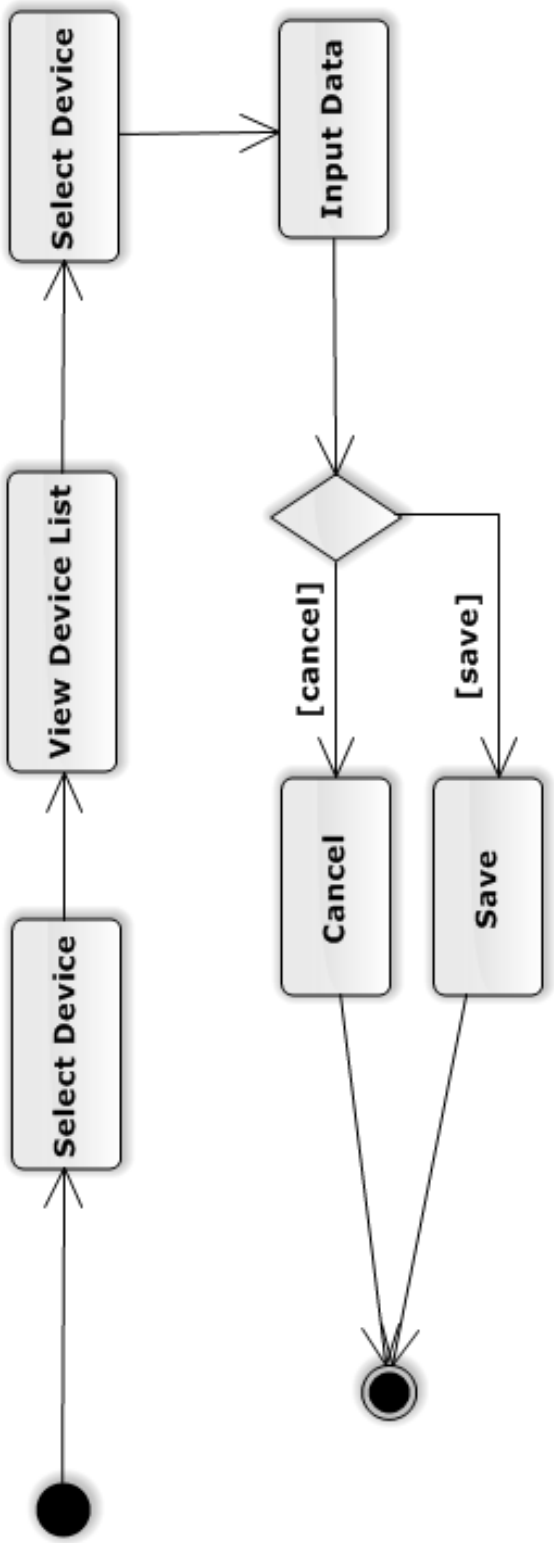

Figure 35 - Input Data Activity Diagram

Manage Sample Data Activity Diagram

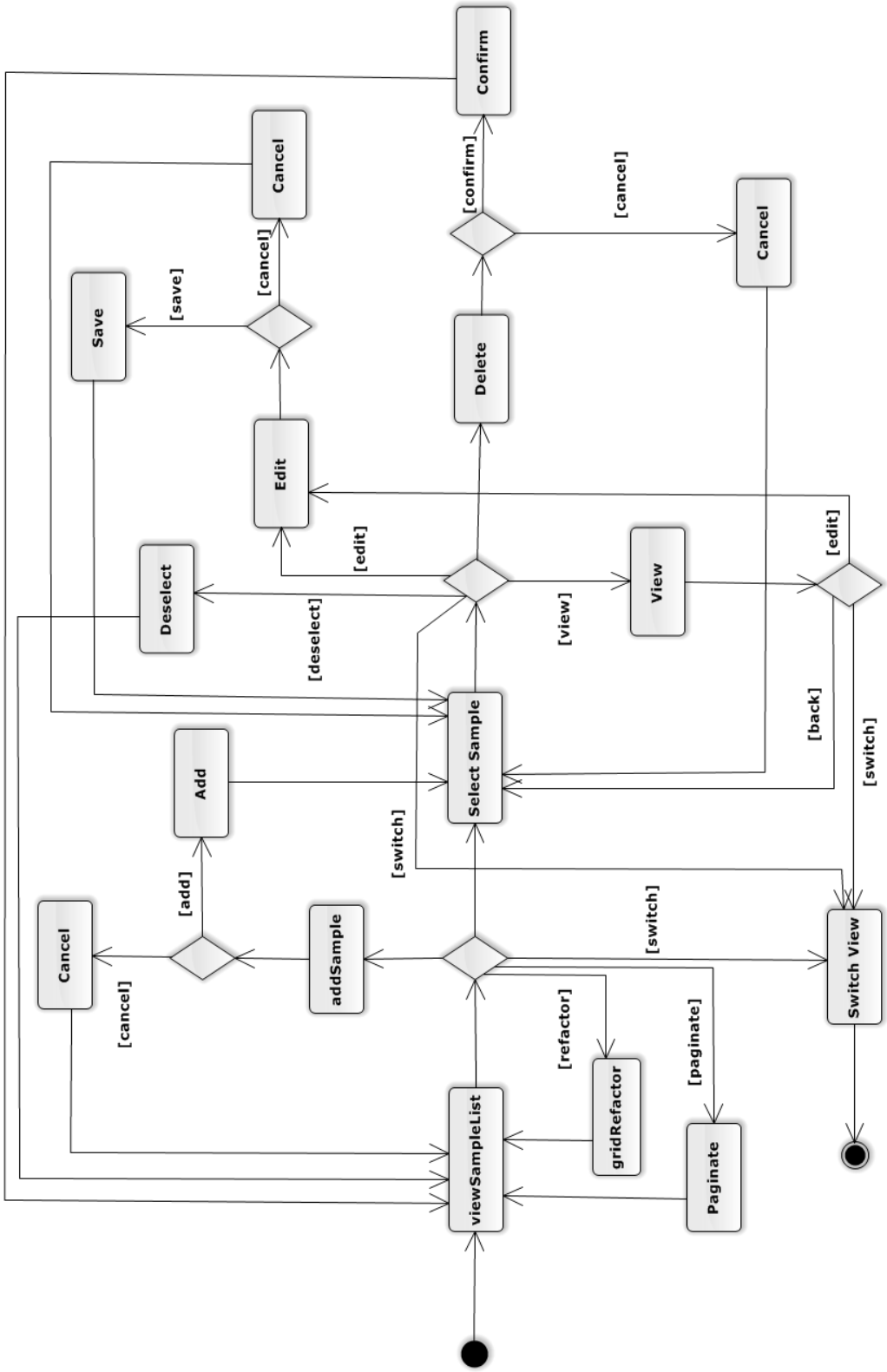

Figure 36 - Manage Sample Data Activity Diagram

Manage Users Activity Diagram

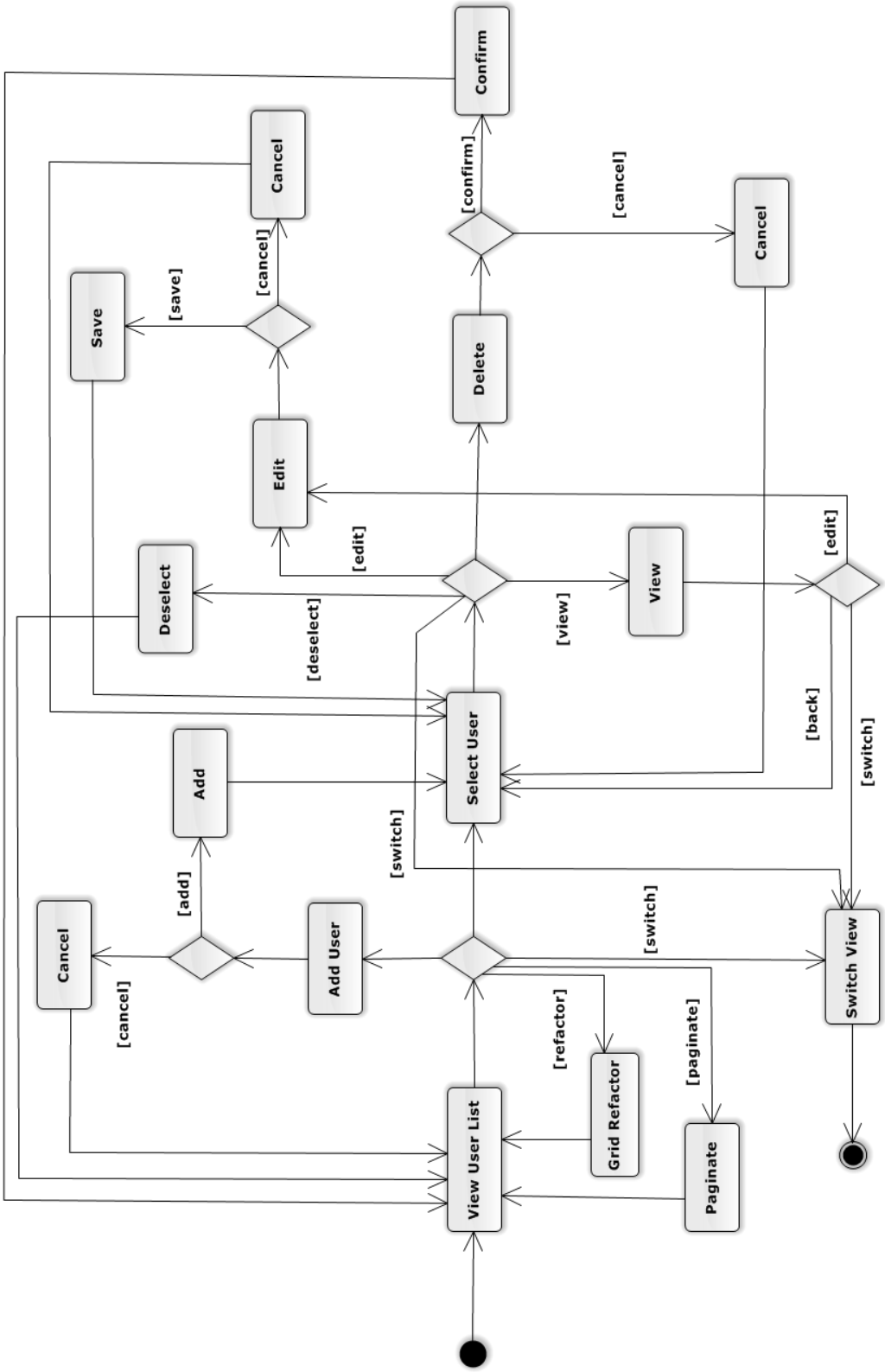

Figure 37 - Manage Users Activity Diagram

Manage Clients Activity Diagram

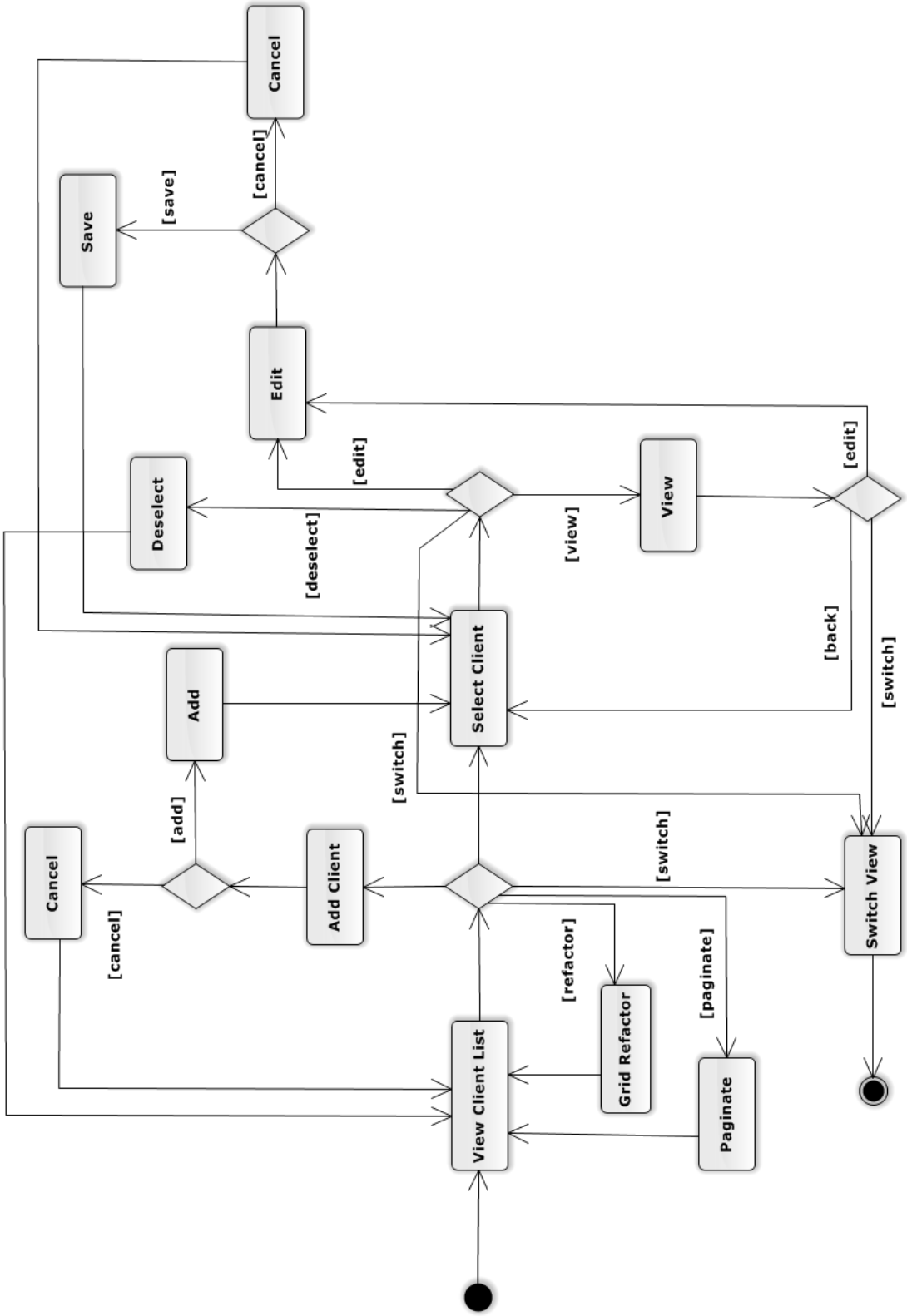

Figure 38 - Manage Clients Activity Diagram

Manage Devices Activity Diagram

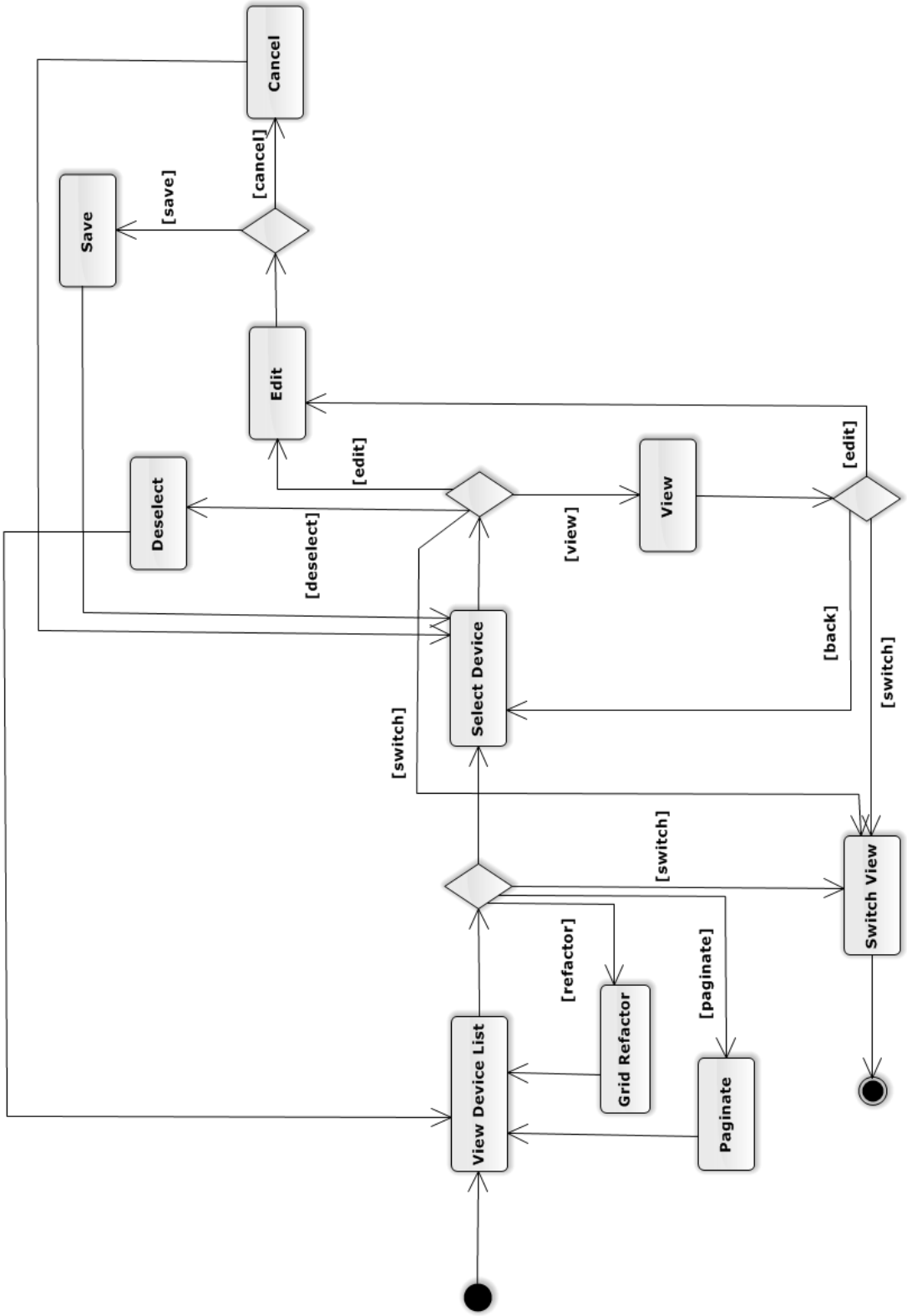

Figure 39 - Manage Devices Activity Diagram

Manage Projects Activity Diagram

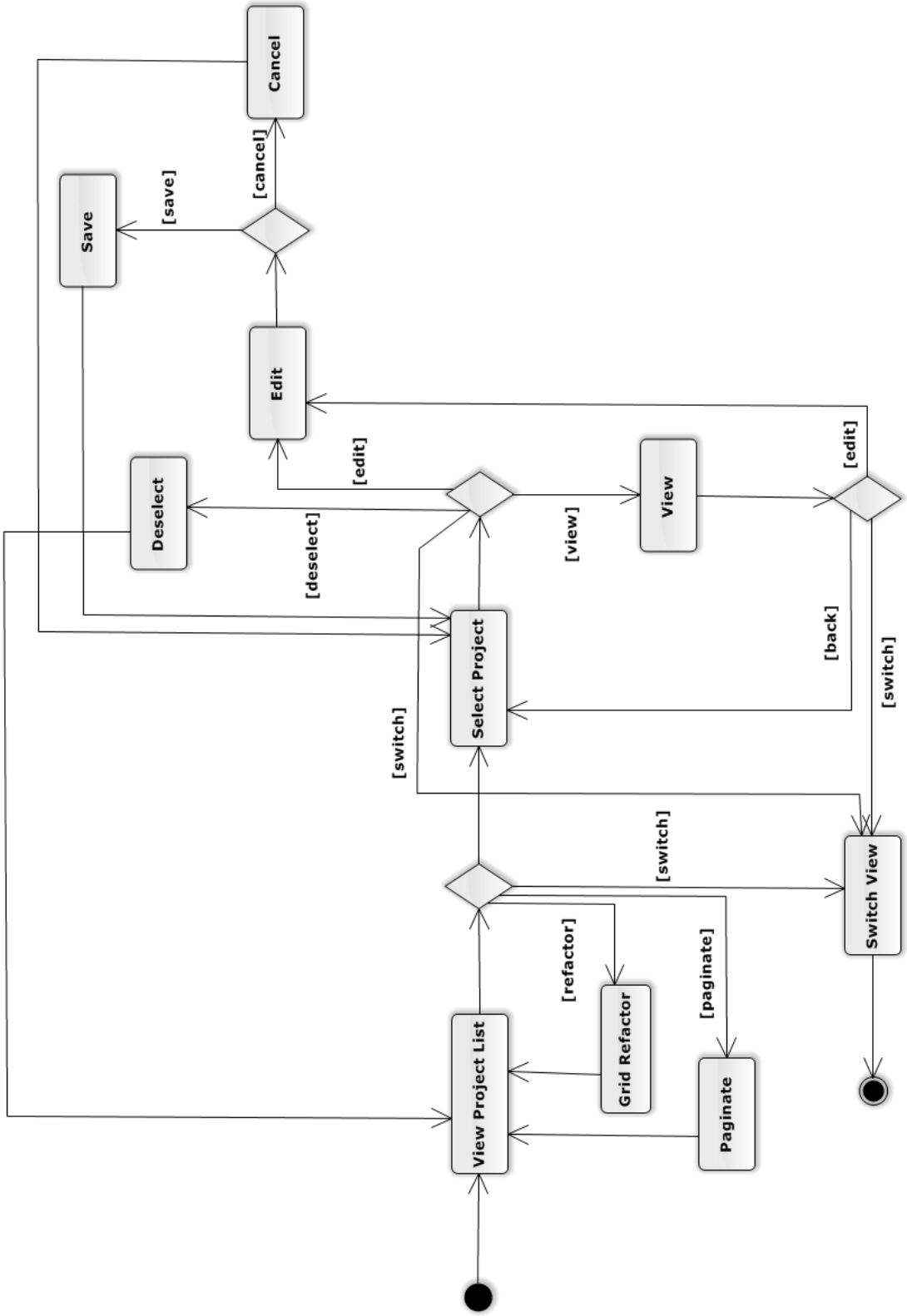

Figure 40 - Manage Projects Activity Diagram

Interaction Model

Interface Description

The front end, user interface, will be designed to mimic modern web page design techniques. The web application will be a static web page with the main content on the page changing depending on the interaction action made by the user.

In accordance with modern user interface design, there will be minimal to no pop-ups that appear on the page. Pop-ups will only be used in situations that require multi-level confirmation for an action, e.g. when the user wishes to delete a project from the database. For general notifications, e.g. form has been saved, changes have been discarded, new data needs be to assigned to a project, etc. These notifications will appear both as a temporary notification on the screen as well as an alternate section of the application that will track all notifications.

The application will be designed primarily to be used on a standard desktop, mouse and keyboard. As well the user interface will be designed to operate as expected on tablet devices (screen width greater or equal to 768 px[5]); the application will not be designed to operate on cellular devices.

The application standards will include (Note: all size specifications are stated assuming a screen width of 1900 px, i.e. the screen width of the development machines):

All headers will use Open Sans Font obtained using Google Fonts. All headings on the page will use be a heading level 1 with the default size set as 40px. Any additional headings that are needed in the application will use a heading level 2. The default font size of these headings will be 36 px. Any sub sections that are not part of the main heading's content, but support or provide further information will use the <small> tag and use default HTML5 styling.

All font in the application will use Lora font obtained using Google Fonts. Default font size will be 16 px.

The standard font colors can be found at this link[6] (the URL can be found at the bottom of this page).

[5] pixels
[6] http://tinyurl.com/oxh2a2e

Web Application Interaction Model Summary

This section will provide a brief summary of the expected functionality each screen that appear on the proceeding pages. Each of these pages are designed to showcase the expected functionality of the application and as such the font, size, colors, layout and wording are subject to change as production of the application commences.

- **Login**
  This screen allows the user to log into the application. The picture in the background will be 1 of many pictures that represent the ARIS lab and will be different each time the login page is accessed.
- **Manage Data**
  This screen showcases the standard grid that will be used to display data. Some examples of where this grid will be used includes:
  - Manage Projects
  - Manage Devices
  - All admin pages
  - Manage Samples

  Key notes about this grid is it will have the ability to sort information based off any columns, filters can be applied to specifically view and find information within the grid.
- **Manually Input Data**
  This screen showcases the ability the user will have to manually input data into the application. The specific input fields will change depending on the preceding screen and in the case of *Devices* the input fields will populate depending on the specific device type selected.
- **Project/Device General Information**
  This screen showcases the overview screen showing all the information for the *Project* or *Device* selected.
- **Project/Device Sample Information**
  This screen showcases the list of all *Samples* associated with the selected *Project* or *Device*.
- **Modal Confirmation**
  This screen showcases how the application will handle pop-up windows. Due to the application specifications, this is the pop-up style that will be used. The grey background will be used to hide the contents of the page, forcing the user to make a selection to either save changes or cancel them.
- **Notifications**
  This screen showcases the notification window that will appear when the user clicks on the notification symbol (in the case of these screens, the bell symbol).

Web Application Interaction Models

Login

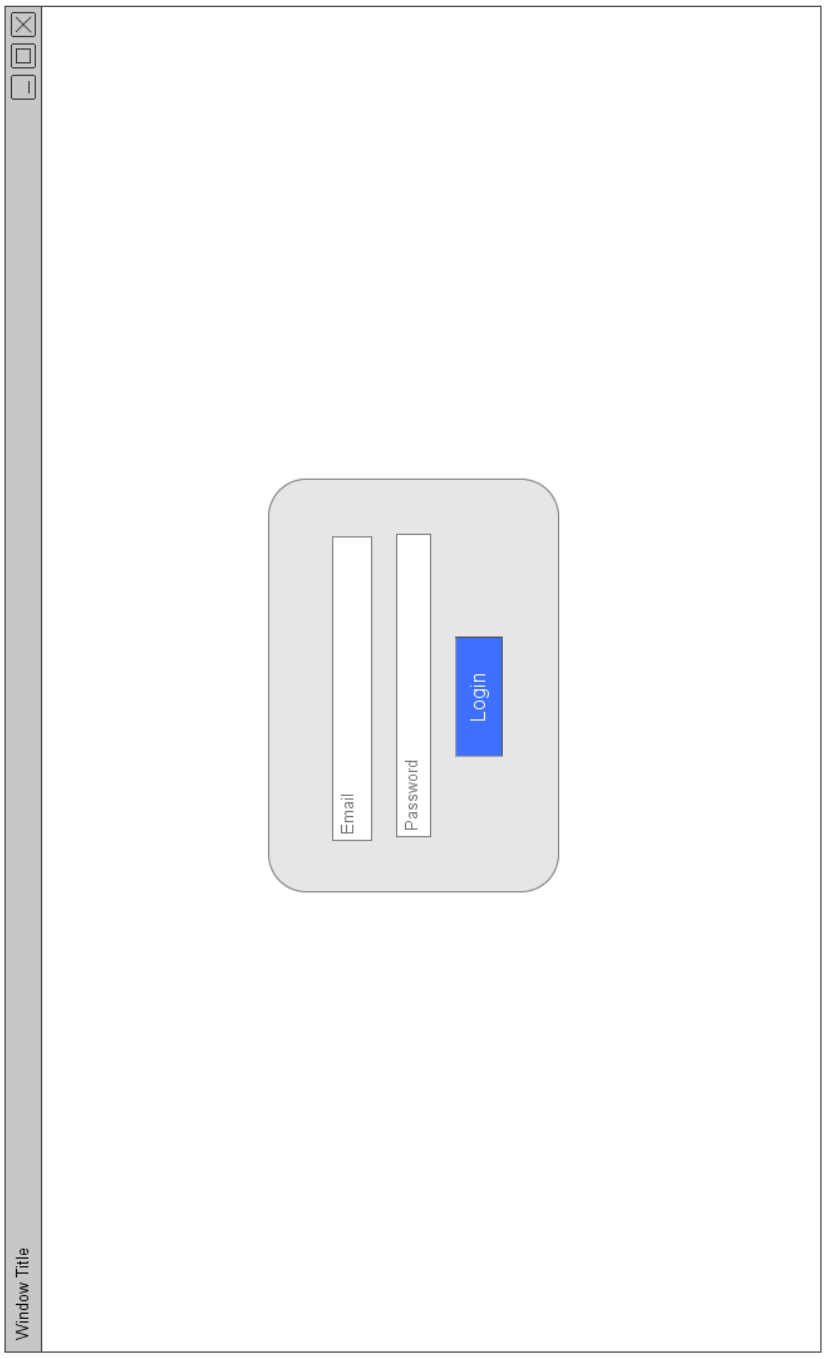

Figure 41 - Login Screen

Manage Data

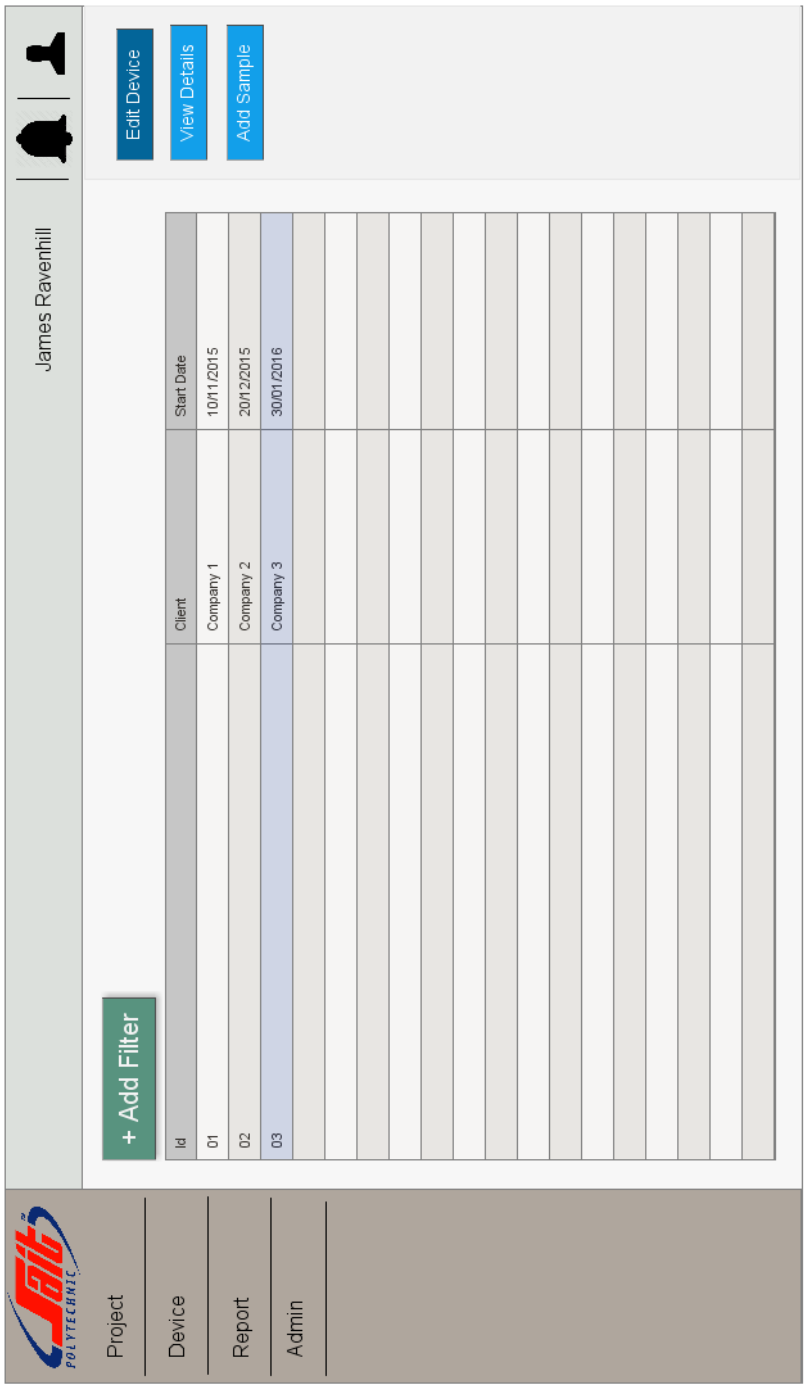

Figure 42 - Manage Data Screen

Manually Input Data

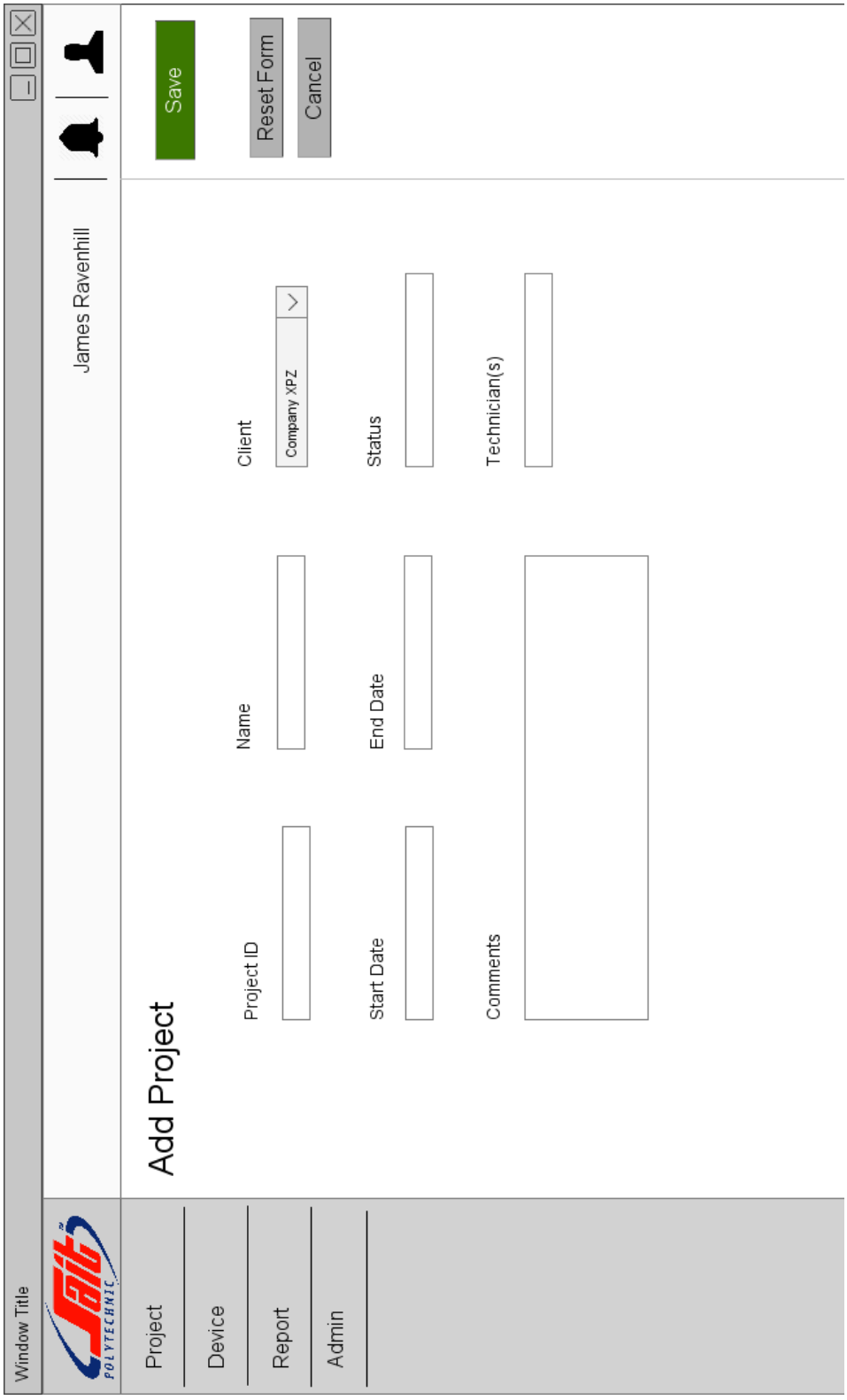

Figure 43 - Manually Input Data Screen

Project/Device General Information

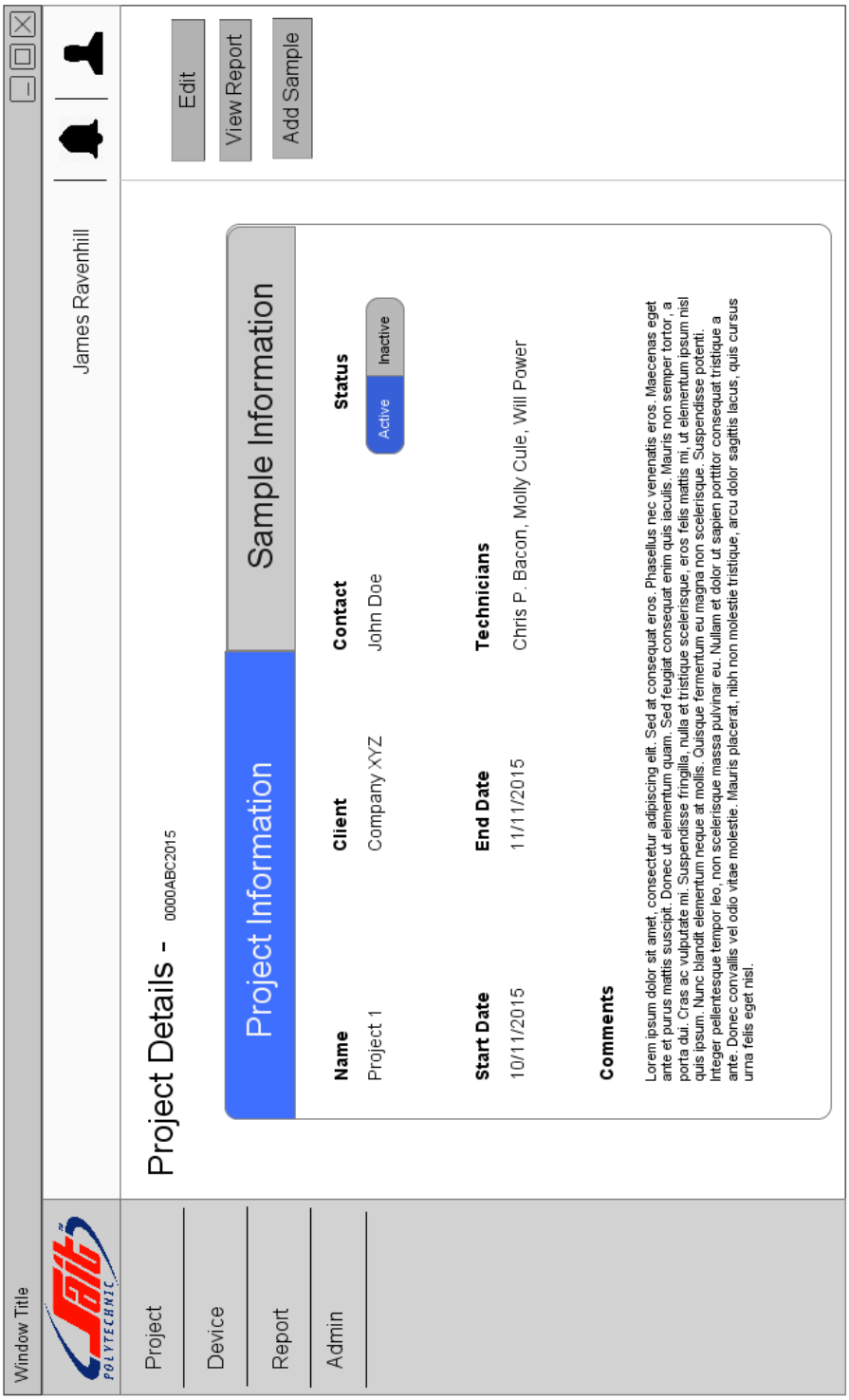

Figure 44 - Project/Device General Information Screen

Project/Device Sample Information

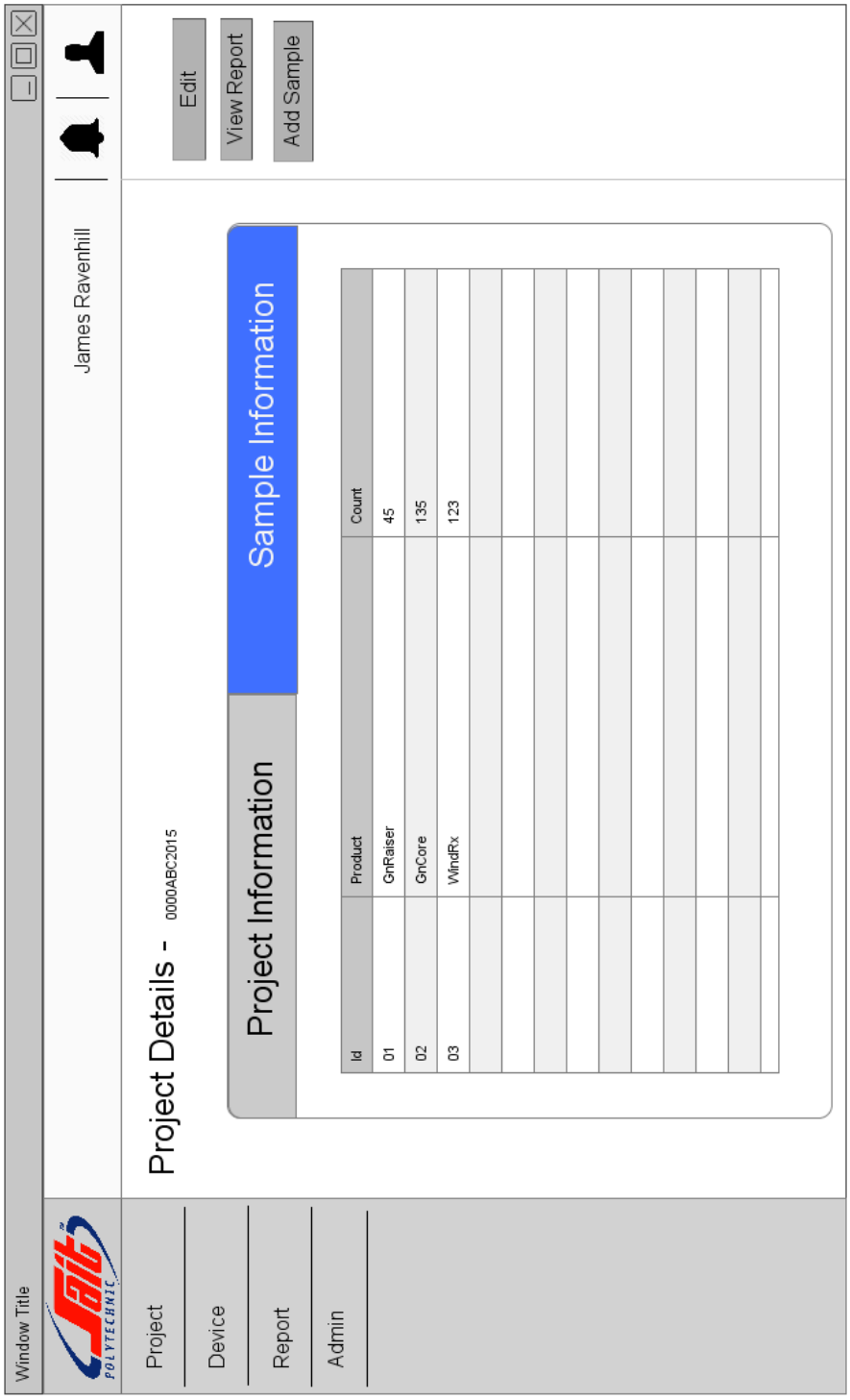

Figure 45 - Project/Device Sample Information Screen

Modal Confirmation

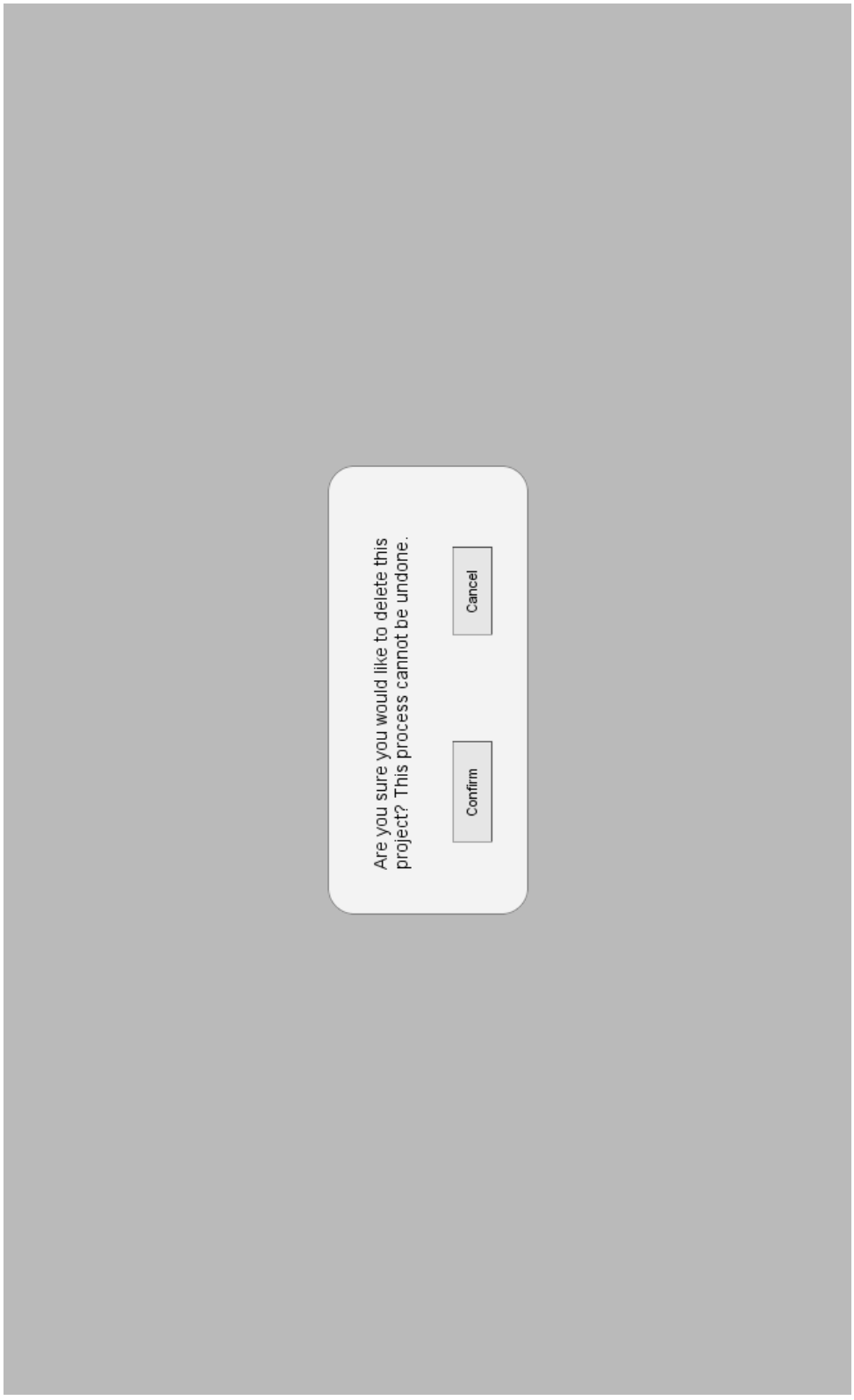

Figure 46 - Modal Confirmation Screen

Notifications

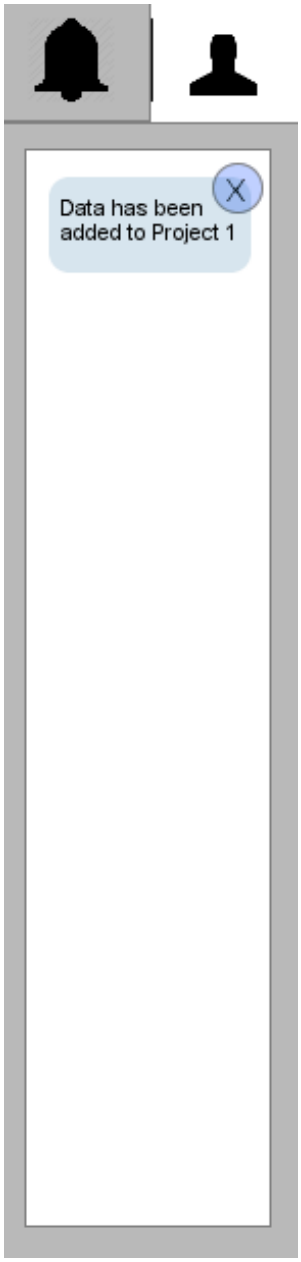

Figure 47 - Notifications Screen

Hardware Architecture

Deployment Diagram

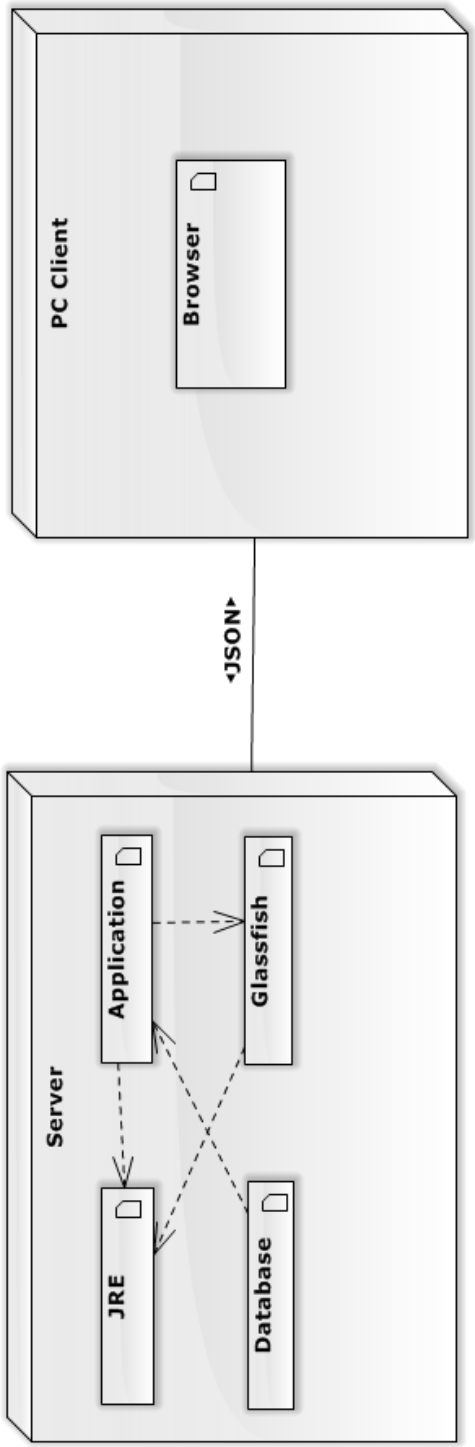

Figure 48 - Deployment Diagram

Communication Mechanisms Diagram

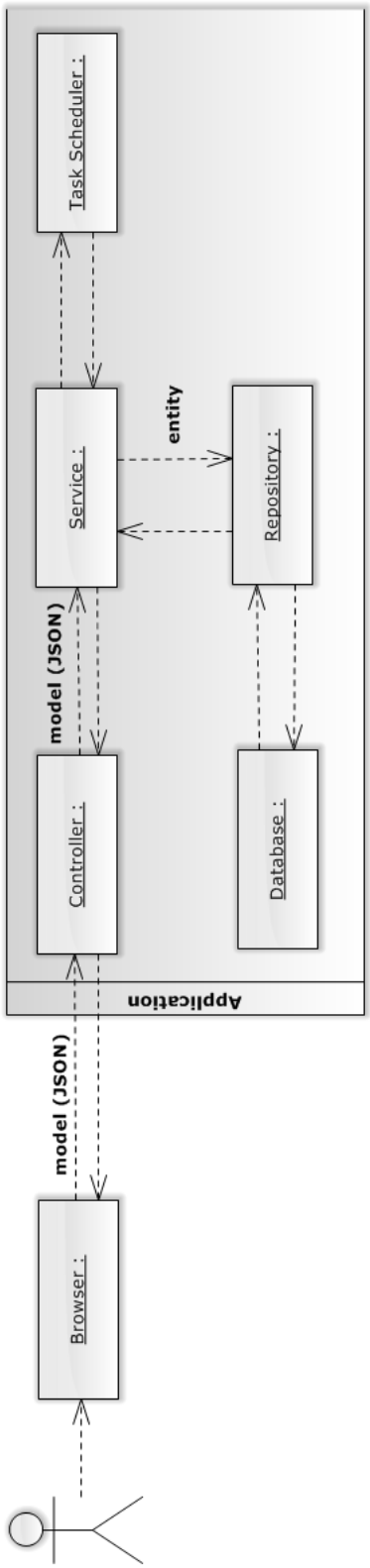

Figure 49 - Communication Mechanisms Diagram

Hardware Platform

Development

The application will be developed primarily on Toshiba Tercra laptops (Windows based).

Production

The application will operate on Windows based computers, primary Dell, Lenovo and HP models.

Software Platform

Development

The application will be developed using the Intellij IDE manufactured by JetBrains S.R.O. We will be using a private BitBucket repository to manage and share application code using Git. We will using SourceTree, manufactured by Atlassian, to manage our Git repository and will be using Tortoise SVN to effectively manage any merge conflicts in our Git commits. We will be using MySQL Workspace, manufactured by Oracle, to view our database. We will be using HipChat, manufactured by Atlassian, to track our commits and changes to the Git repository.

Production

The application will operate on a Linux server, hosted by the ARIS RAD Lab. The computers that will be connecting to the application will use a Windows 7 operating system and be using the Google Chrome web browser.

Data Dictionary

Frameworks

Gradle

Gradle is an advanced build system as well as an advanced build toolkit allowing to create custom build logic through plugins. Here are some of its features that made us choose Gradle [8]:

- Domain Specific Language (DSL) based on Groovy, used to describe and manipulate the build logic
- Build files are Groovy based and allow mixing of declarative elements through the DSL and using code to manipulate the DSL elements to provide custom logic.
- Built-in dependency management through Maven and/or Ivy.
- Very flexible. Allows using best practices but doesn't force its own way of doing things.
- Plugins can expose their own DSL and their own API for build files to use.
- Good Tooling API allowing IDE integration.

Spring Boot

Spring Boot makes it easy to create stand-alone, production-grade Spring based applications that you can "just run". We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration. [9]

Data JPA

Spring Data JPA, part of the larger Spring Data family, makes it easy to easily implement JPA based repositories. This module deals with enhanced support for JPA based data access layers. It makes it easier to build Spring-powered applications that use data access technologies. [10]

Data Rest

Spring Data REST is part of the umbrella Spring Data project and makes it easy to build hypermedia-driven REST web services on top of Spring Data repositories. [11]

Security

Spring Security is a framework that focuses on providing both authentication and authorization to Java applications. Like all Spring projects, the real power of Spring Security is found in how easily it can be extended to meet custom requirements. [13]

JavaScript

Font Awesome

Font Awesome gives you scalable vector icons that can instantly be customized — size, color, drop shadow, and anything that can be done with the power of CSS. [16]

JQuery

JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, JQuery has changed the way that millions of people write JavaScript. [17]

AngularJS

AngularJS lets you write client-side web applications as if you had a smarter browser. It lets you use good old HTML (or HAML, Jade and friends!) as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. It automatically synchronizes data from your UI (view) with your JavaScript objects (model) through 2-way data binding. To help you structure your application better and make it easy to test, AngularJS teaches the browser how to do dependency injection and inversion of control. [18]

Angular Route

Use Angular Route to enable URL routing to your application. The Angular Route module supports URL management via both hashbang and HTML5 pushState. [19]

Angular Resource

Use the Angular Resource module when querying and posting data to a REST API. [19]

System Administration

Security

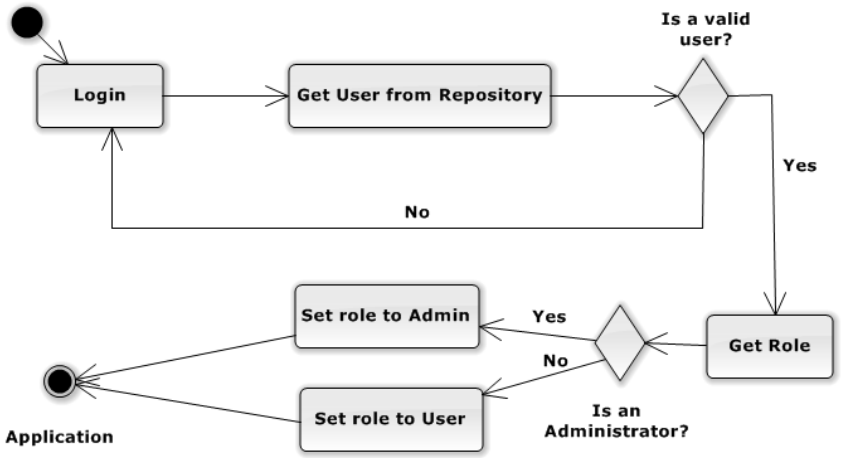

Figure 50 - Security Activity Diagram

Operations

The system will startup automatically on a server restart. Glassfish will start automatically on a system restart and MySQL will start automatically shortly after Glassfish. After the MySQL is started the application will automatically startup. The client can start and stop the server as needed as well which will require a manual restart of the application.

On a shutdown, MySQL and Glassfish will receive the terminate command from the Operating System and shutdown along with the system. No manual interaction will be required unless something breaks

MySQL will install its updates automatically as they are released, but Glassfish will not install any updates, this will require doing it manually, but this should not need to be updated in the foreseeable future.

Backup and Restore

Using the tool MySqlDump, the application will perform a full backup of the entire database every night at 2:15 am. Files will be saved according to their respective dates (day, month, year). To restore the data for the required day will be a restoral page in the application. The page will ask for the day, month and year to be backed up to. The application will run the following line of code to retrieve the correct file:

```
15 2 * * * root mysqldump -u root -pPASSWORD --all-databases |
gzip > /mnt/disk2/database_`data ' %m-%d-%Y'`.sql.gz
```

Data Archival

Our client only needs to see completed projects from the past 3 years. After 3 years, the completed projects will be archived. Archiving will be done by exporting the entire project, using Java, to CSV, where it will be stored to a user specified location. Archived projects can be imported into the database, if needed, using SQL Loader. This can only be done if the provided the database schema has not changed dramatically since the project was originally archived.

Project Management

Schedule

Major Milestones

- Requirements document to be submitted Oct 5, 2015
- Requirements analysis document to be submitted Nov 1, 2015
- Requirements design document to be submitted Dec 11, 2015
- Final requirements document to be submitted Jan 25, 2016
- Client information meeting will be on either Feb 2 or 3, 2016
- Technical information meeting will be on either March 1 or 2, 2016
- Completed project and implementation will be on April 15, 2016

Meeting Schedule

Meet with client bi-weekly on Fridays at 3 pm in ARIS RAD Lab, room MD315 at SAIT during September 2015 through December 2015. From January 2016 until April 2016 the time of the meetings will change to Wednesday 10:30 am; location will remain the same. In the case we are unable to meet in MD315, meetings will take place in CA416 on the SATI campus.

Work Breakdown Gnatt Chart

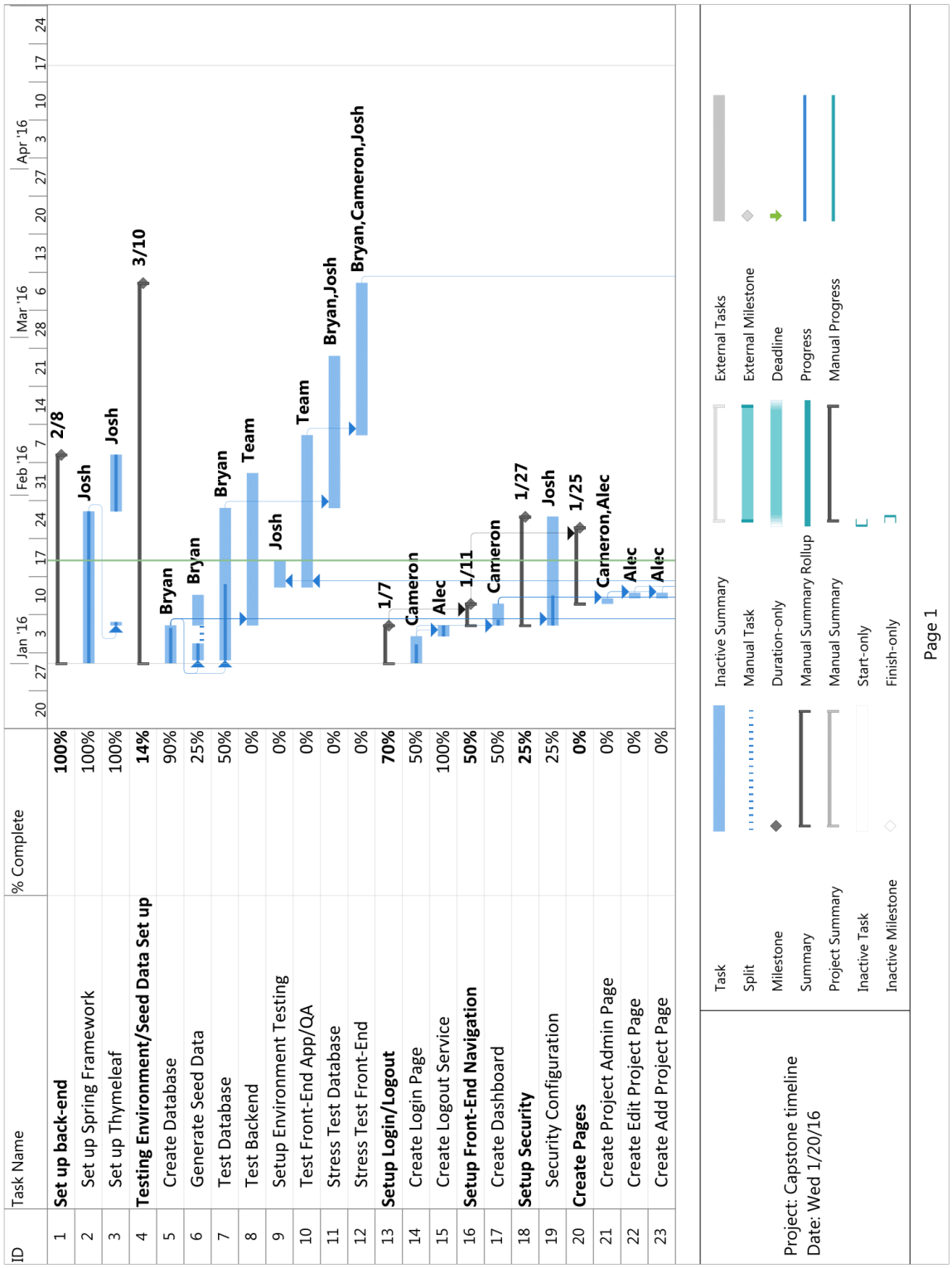

Figure 51 - Gnatt Chart Page 1

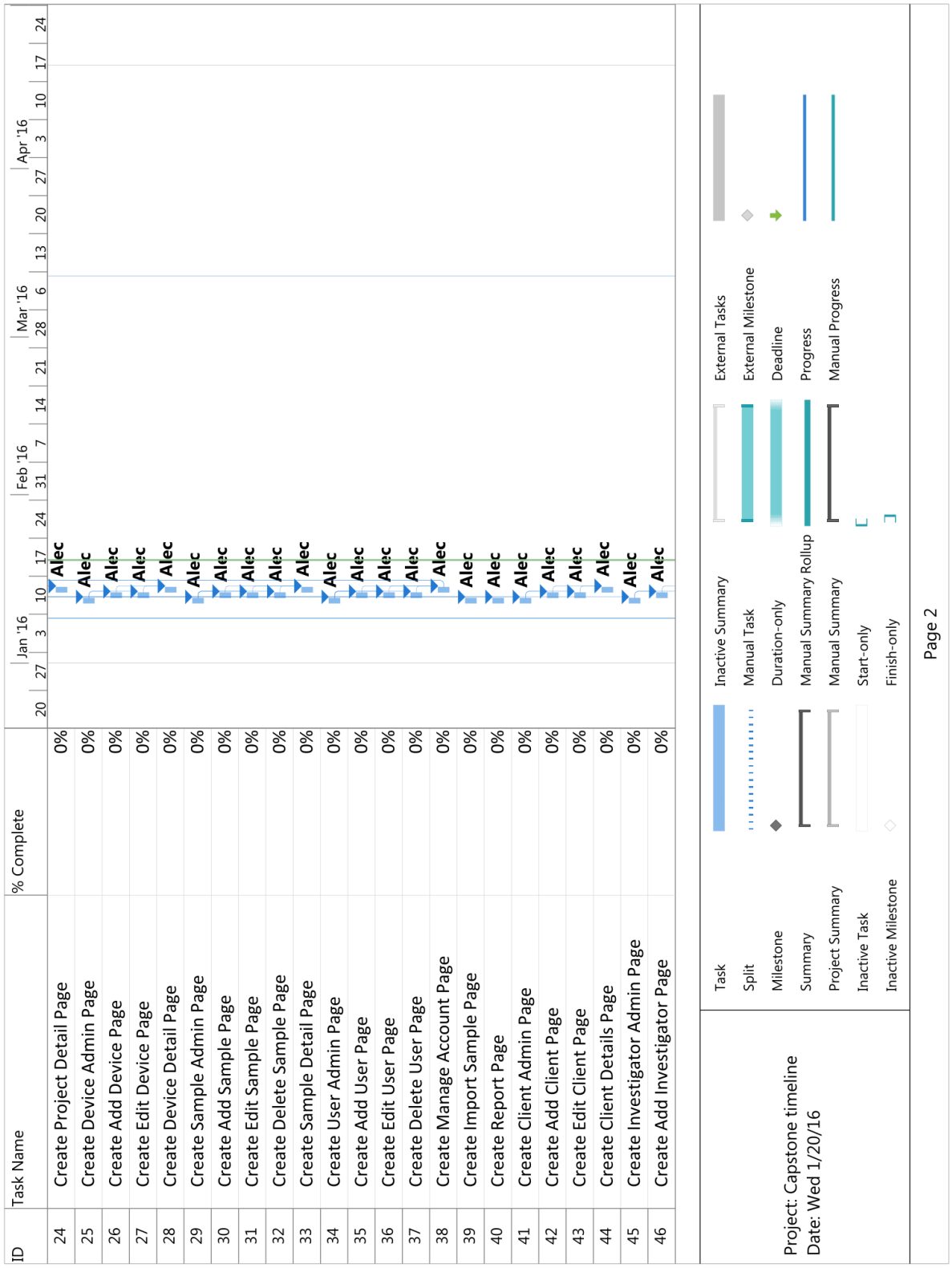

Figure 52 - Gnatt Chart Page 2

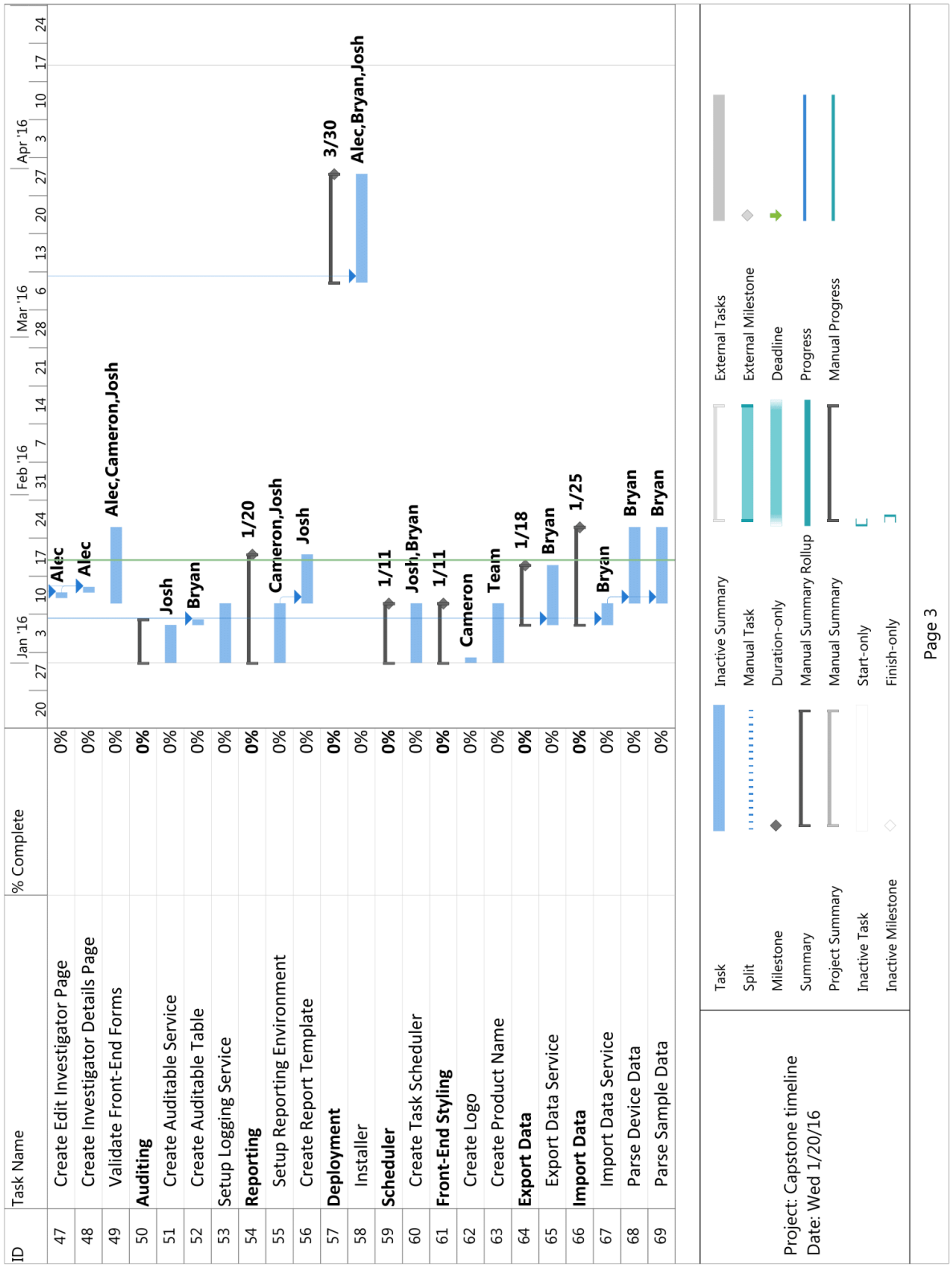

Figure 53 - Gnatt Chart Page 3

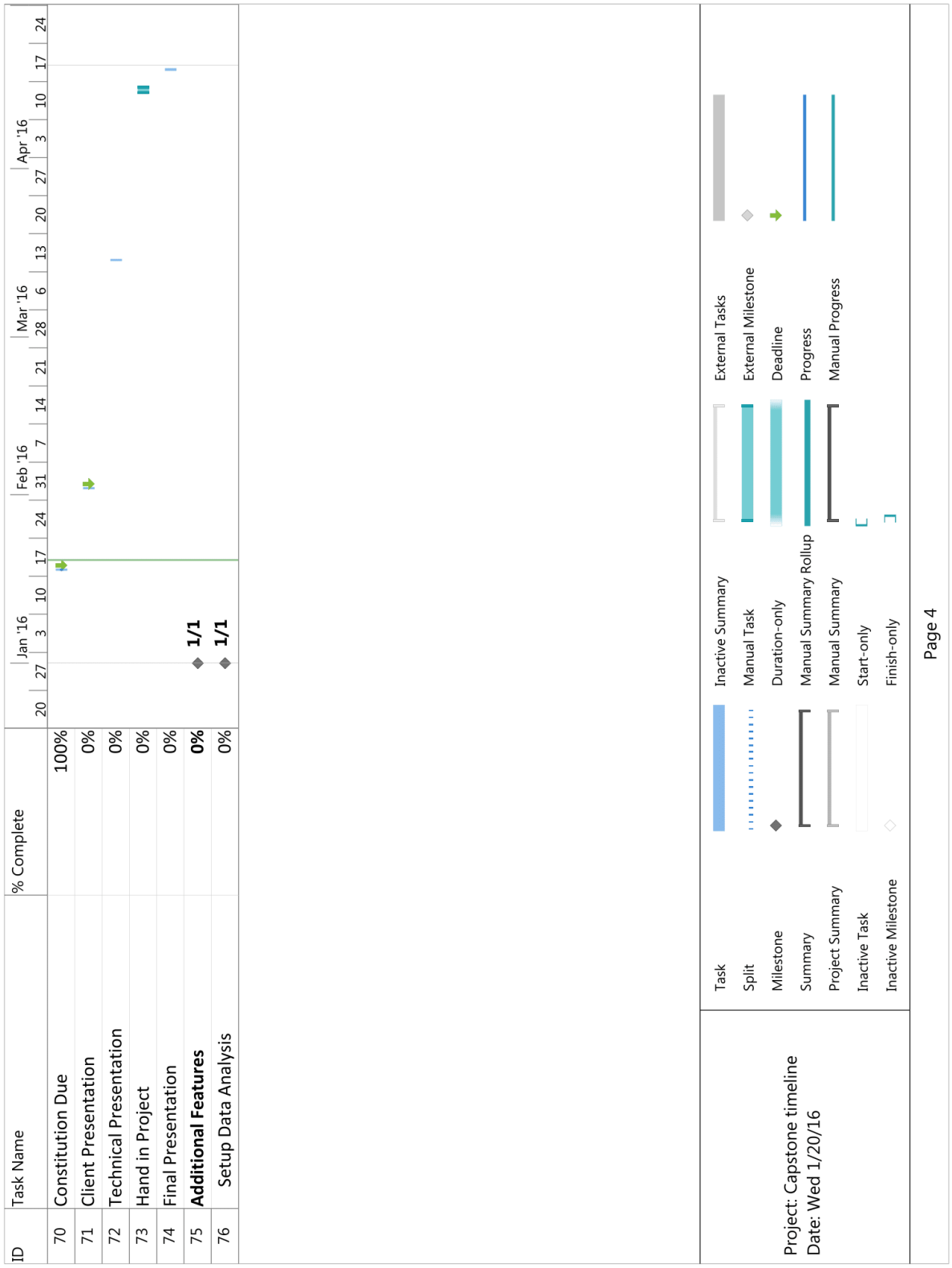

Figure 54 - Gnatt Chart Page 4

Team Configuration

Below is a table outlining the development team and their primary roles:

Team Member	Roles
Alec Wassill	- Build and implement all client-side pages - Ensure communication between client-side and server - Implement client-side validation - QA
Bryan Martinez	- Design and implement database - Test all database entities - Create method to automatically import, parse and add new data to the system - Reporting - Create (un)installation procedures
Cameron McNaught	- Design client-side appearance - Build and maintain various custom components - Manage all documentation - Resolve most issues regarding AngularJS
Josh Lynn	- Setup and maintain base frameworks - Implement security - Designed and implemented methods to filter and sort grid content - Resolved all routing issues - Resolved some issues regarding AngularJS - Created generic CRUD - Created all services

Project Standards and Procedures

For the application the following tools will be utilized, including versions if applicable.

Resource	Version
Server	
MySQL	5.7.10
MySQL Dump	4.5.4
Backend	
Java JDK[7]	1.8_60
Spring Boot	1.3.1.RELEASE
Spring Boot Data JPA	(see above)
Spring Boot Data Rest	(see above)
Spring Boot Web	(see above)
Spring Boot Security	(see above)
Spring Boot Test	(see above)
Spring Boot Devtools	(see above)
Spring Boot Model Mapper	1.1.0
Mockito	1.10.19
HyperSQL Database	2.3.3
Dynamic Reports	4.0.1
Front end	
AngularJS	1.5
Angular Material	1.1.0
LESS	2.5.3
Font-Awesome	4.5.0
JQuery	1.10
Other	
Intellij	14.1.15
BitBucket	N/A
SourceTree	N/A
Gradle	2.9
Git	N/A
HipChat	N/A

[7] Java Development Kit

Glossary

ARIS	Applied Research and Innovation Services. More information about ARIS can be found here: http://www.sait.ca/research-and-innovation/about-aris.php
BitBucket	BitBucket provides Git and Mercurial hosting for teams. Host your code online in as many public and private repositories as you want. Free five-user tier accounts! Manage your projects with confidence with built-in issue trackers, wikis, code comments, and pull requests. [20]
CSS	Cascading Style Sheets. Used to format the layout of web pages. [22]
CSV	Comma Separated Values. A method of organizing data separated by commas [,]. Files can be distinguished as a Comma Separated Values file by the extension .csv.
Excel Document	A file created using the Microsoft Office product Microsoft Excel.
IDE	Integrated Development Environment. A development tool used to write and organize software code.
Intellij	A develop environment developed by JetBrains used to develop Java based projects.
MySQL	MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MySQL is an essential part of almost every open source PHP application. [23]
Java JDK	Java Development Environment.
Spring MVC	The Spring web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between model, view and controller. [24]

Index

U

W

Works Cited

[1] SAIT, "Environmental Technologies," SAIT, 2015. [Online]. Available: http://www.sait.ca/research-and-innovation/environmental-technologies.php. [Accessed 28 September 2015].

[2] Android, "Gradle Plugin User Guide," 2015. [Online]. Available: http://tools.android.com/tech-docs/new-build-system/user-guide#TOC-Introduction. [Accessed 6 Dec 2015].

[3] Pivotal Software Inc, "Spring Boot," Pivitoal Software Inc, 2015. [Online]. Available: http://projects.spring.io/spring-boot/. [Accessed 6 Dec 2015].

[4] Pivotal Software Inc, "Spring Boot," Pivotal Software, 2015. [Online]. Available: http://projects.spring.io/spring-data-jpa/. [Accessed 6 Dec 2015].

[5] Pivotal Software Inc, "Spring Data REST," Pivotal Software Inc, 2015. [Online]. Available: http://projects.spring.io/spring-data-rest/. [Accessed 6 Dec 2015].

[6] Pivotal Software Inc., "Spring Security," 2015. [Online]. Available: http://projects.spring.io/spring-security/. [Accessed 6 Dec 2015].

[7] D. Gandy, "Font Awesome, the iconic font and CSS toolkit," 2015. [Online]. Available: https://fortawesome.github.io/Font-Awesome/. [Accessed 6 Dec 2015].

[8] The JQuery Foundation, "JQuery," 2015. [Online]. Available: https://jquery.com/. [Accessed 6 Dec 2015].

[9] Google, "angular/angular.js," 2015. [Online]. Available: https://github.com/angular/angular.js. [Accessed 6 Dec 2015].

[10] Google, "Angular JS: API Reference," 2015. [Online]. Available: https://docs.angularjs.org/api. [Accessed 6 Dec 2015].

[11] Team Koding, "What is Bitbucket?," 1 5 2015. [Online]. Available: http://learn.koding.com/guides/using-bitbucket-on-koding/. [Accessed 2 11 2015].

[12] P. Christensson, "CSS Definition," 2006. [Online]. Available: http://techterms.com/definition/css. [Accessed 1 Oct 2015].

[13] Site Ground, "My SQL Tutorial," 2015. [Online]. Available: https://www.siteground.com/tutorials/php-mysql/mysql.htm. [Accessed 2 11 2015].

[14] TutorialsPoint, "Spring MVC Framework Tutorial," 2015. [Online]. Available: http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm. [Accessed 2 11 2015].

[15] Tutorial Republic, "What is Bootstrap 3 - A Brief Introduction," 2015. [Online]. Available: http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-introduction.php. [Accessed 2 11 2015].

[16] THYMELEAF Team, "Thymeleaf: java XML/XHTML/HTML5 template engine," 2015. [Online]. Available: http://www.thymeleaf.org/index.html. [Accessed 6 Dec 2015].

[17] MySql, "MySql :: Download Connector/J," Oracle, 2015. [Online]. Available: http://dev.mysql.com/downloads/connector/j/. [Accessed 6 Dec 2015].

[18] Bootstrap, "Bootstrap · The world's most popular mobile-first and responsive front-end framework.," 2015. [Online]. Available: http://getbootstrap.com/. [Accessed 6 Dec 2015].

[19] Jetbrains S.R.O, "Intellij IDEA - The Most Intelligent Java IDE," Jetbrains, 2015. [Online]. Available: https://www.jetbrains.com/idea/. [Accessed 9 Dec 2015].

[20] Atlassian, "Free Mercurial and Git Client for Windows and Mac | Atlassian SourceTree," Atlassian, 2014. [Online]. Available: https://www.sourcetreeapp.com/. [Accessed 9 Dec 2015].

[21] TortoiseSVN, "Home - TortoiseSVN," TortoiseSVN Team, 2015. [Online]. Available: https://tortoisesvn.net/about.html. [Accessed 9 Dec 2015].

[22] Atlassian, "Bitbucket - The Git solution for professional teams," Atlassian, 2015. [Online]. Available: https://bitbucket.org/. [Accessed 9 Dec 2015].

[23] MySQL, "MySQL :: MySQL Workbench," Oracle, 2015. [Online]. Available: https://www.mysql.com/products/workbench/. [Accessed 9 Dec 2015].

[24] Atlassian, "HipChat - Private group chat, video chat, instant messaging for teams | HipChat," Atlassian, 2015. [Online]. Available: https://www.hipchat.com/. [Accessed 9 Dec 2015].

Addendum

Conceptual Entity Relationship Diagram

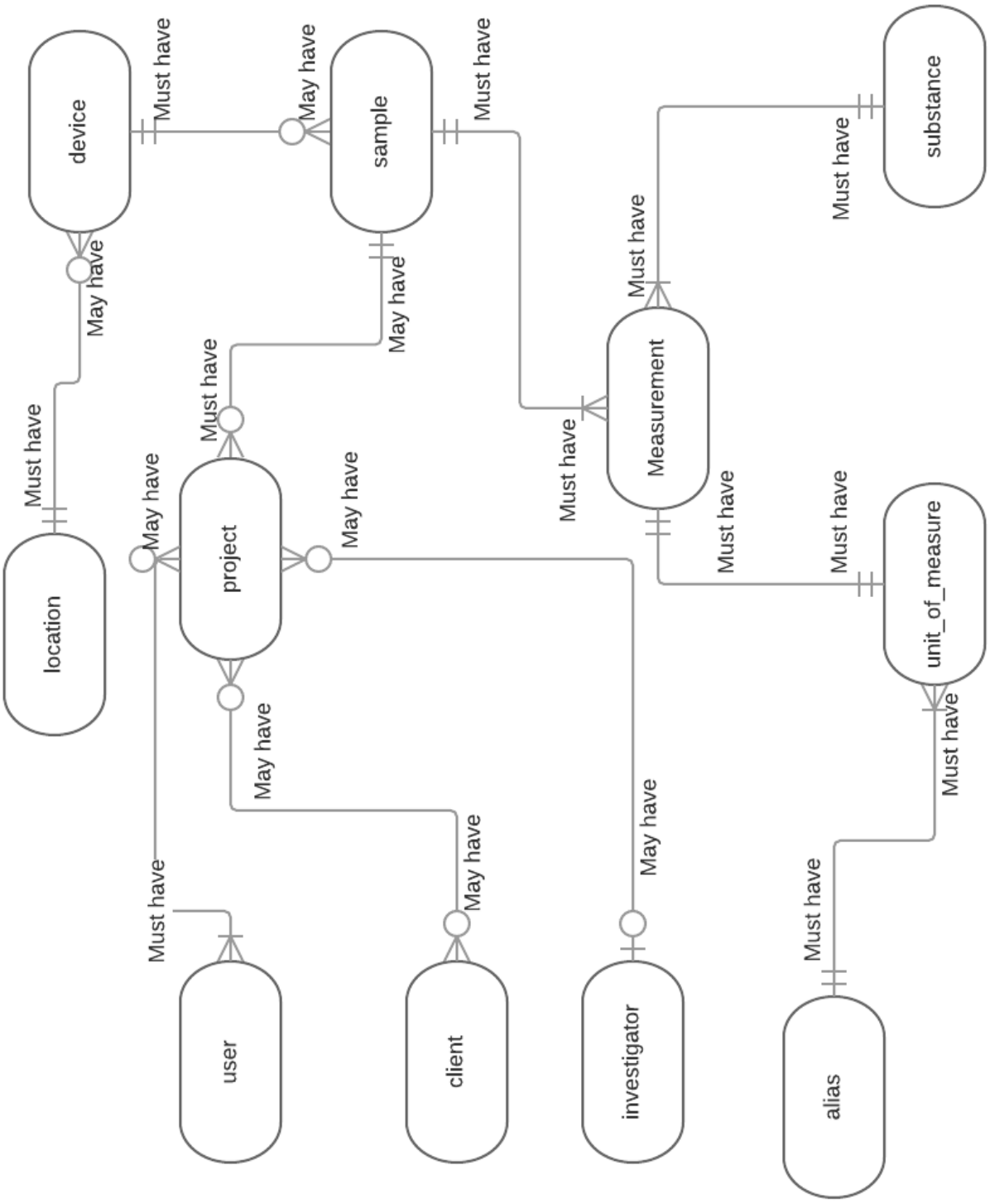

Figure 55 - Conceptual Entity Relationship Diagram