 Department of Computer Science and Engineering  
The University of Texas at Arlington

Aegle

Outreach Inventory System

Team Members:   
*Joseph Finnegan*

*Karla Hernandez*

*Joe Martinez*

*Carlos Torres*

#### Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Revision Date** | **Description** | **Rationale** |
| 0.1 | 10/08/2014 | First Draft | Initial draft for OSS system |
|  |  |  |  |

#### Table of Contents

1. Product Concept 8

1.1 Purpose and Use 8

1.2 Intended Audience 9

2. Product Description and Functional Overview 10

2.1 Features and Functions 10

2.2 External Inputs and Outputs 10

2.3 Product Interfaces 12

3. Customer Requirements 18

3.1 Tracking of Items and Crates by System 21

3.2 System Description of Items 21

3.3 System Description of Crates 21

3.4 Search Function for Items and Crates 22

3.5 Locating Item Inside a Crate 22

3.7 Item Management by the administrators 22

3.8 Crate Management by the administrators 23

3.9 Project Management by administrators 23

3.10 System Interaction by administrators 24

3.11 System Interaction by Registered Users 24

3.12 Registration/Login System 24

3.13 Web-Based Accessible Application 25

4. Packaging Requirements 26

4.1 Included Components 26

4.2 Installation Manual 26

4.3 User Manual 26

4.4 Power Supply 27

4.5 Range of RFID Antenna 27

5. Performance Requirements 28

5.1 Check-in/Check-out Latency 28

5.2 Web Interface Response Latency 28

6. Safety Requirements 29

6.1 Electrical Hazard 29

6.2 Signal Interference 29

7. Maintenance and Support Requirements 30

User Manual 30

7.2 Troubleshooting Guide 30

7.3 Source Code Availability & Documentation 31

8. Other Requirements 32

8.1 Security 32

8.2 Connectivity Tolerance 32

9. Acceptance Requirements 33

9.1 Verify User Account Registration 33

9.2 Verify Tracking an Item Functionality 33

9.3 Verify Tracking a Crate Functionality 34

9.4 Verify Search Functionality 34

9.5 Verify Item Management Functionality 34

9.6 Verify Crate Management Functionality 35

9.7 Verify Project Management Functionality 36

10. Use Cases 37

10.1 Registering into the System 37

10.2 Logging into the System 37

10.3 Adding Item, Crate, or Project to the System 38

10.4 Editing Item, Crate, or Project in the System 38

10.5 Deleting Item, Crate, or Project from the System 39

10.6 Searching for Items and Crates 39

10.7 User Registration Confirmation by administrator 39

10.8 Item/Crate Check-out Approval by administrator 40

10.9 Item/Crate Checkout Request by Registered User 40

10.10 Item\Crate Checkout by administrator 40

10.11 Item\Crate Check-in by administrators and Registered Users 41

11. Feasibility Assessment 45

11.1 Scope Analysis 45

11.2 Research 45

11.3 Technical Analysis 46

11.4 Cost Analysis 47

11.5 Resource Analysis 48

11.6 Schedule Analysis 49

12. Future Items 54

12.1 Crate Containment Tracking 54

12.2 Mobile Application 54

12.3 Locating Crate Inside the Storage Room 54

#### List of Figures

Figure Page

Figure 1‑1 Conceptual Diagram 9

Figure 2‑1 Mockup prototype for Home page 12

Figure 2‑2 Mockup prototype for Login / Register page 13

Figure 2‑3 Mockup prototype for Search page 14

Figure 2‑4 Mockup Prototyoe for User Management page 15

Figure 2‑5 Mockup prototype for Store Management page 16

Figure 2‑6 Mockup prototype for Search Results page 17

Figure 10‑1 Login and Registration System 42

Figure 10‑2 Inventory check in /out system 43

Figure 10‑3 Inventory Management 44

#### List of Tables

Table Page

**Table 2‑1** Table of inputs and outputs 11

**Table 11‑1:** Preliminary Cost Analysis 47

**Table 11‑2**: Function Point Table Breakdown 49

**Table 11‑3**: Influence Multipliers 50

**Table 11‑4**: Jones First Order Estimation 51

**Table 111‑5**: Simplified Hybrid Approach 52

# Product Concept

The following section describes the purpose, use, and intended user audience for the Outreach Storage System. The purpose of OSS is to manage project inventory for Dr. Tiernan the Outreach Coordinator at UTA. As Outreach Coordinator, Dr. Tiernan is responsible for organizing events and other activities to promote the College of Engineering to college bound students.

Many of these activities require different types of materials and tools to complete, when not in use these materials are stored in a large room in Nedderman Hall. This storeroom until now has been organized by pen and paper, over time this had led to it becoming slowly disorganized as those with access to the storeroom have added and removed items haphazardly and without updating inventory counts. Because of this there is no precise accounting for what items and exactly how many of those items are in the storeroom at any one time.

This system will save time and money by keeping track of inventory and allowing for more efficient use of resources.

## 1.1 Purpose and Use

Managing product inventory is a problem that has been with human societies since the merchants of Sumeria. In the past this required meticulous record keeping and regular auditing to ensure accurate inventory records. OSS seeks to automate these processes and add new tasks specific to our customer’s needs.

This system will use RFID technology to automate previously manual tasks. Inventory records will be updated as items are added or removed from storage crates, and their location within the storeroom will be recorded. OSS will also organize inventory into user defined projects, such as constructing a self-supporting tower made of dry spaghetti noodles, or categories, such as tools or crafts.

The goal of OSS will be to save the user the time and effort needed to find an item within a storeroom and to more effectively manage resources. By keeping accurate inventory counts, existing product can be used more efficiently and waste can be reduced.

## 1.2 Intended Audience

The target audience for this system is Dr. Tiernan, the Outreach Coordinator at UTA, and her assistants. The following conceptual diagram illustrates in a high level overview how the OSS will operate.

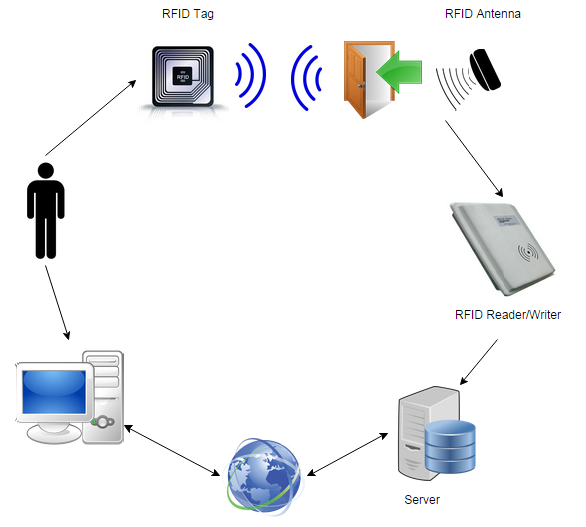


Figure ‑ Conceptual Diagram

# Product Description and Functional Overview

The following section provides the reader with an overview of the Outreach Storage System. The primary operational aspects of the product, from the perspectives of the end users, maintainers, and administrators, are defined here. The key features and functions found in OSS as well as critical user interactions and user interfaces are described in detail.

## 2.1 Features and Functions

OSS will consist of a server, RFID tags, and RFID sensors. Every item that is to be tracked by the system will have an RFID tag attached to it. RFID sensors will read these tags as items are added and removed from the storeroom, in this way keeping an accurate record of the total inventory.

Each item will be assigned to a general category that describes its function, such as electronics, crafts, etc. Each item will also be assigned to a crate; which will store the items inside the storeroom, these crates can themselves be assigned to categories. Finally, items may also be assigned to user-defined projects of which they are a component of.

## 2.2 External Inputs and Outputs

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Use** |
| Home Page | Output | Displays the web application’s Home Page | Portal through which the user will either Log In or Register an account |
| Register Account | Input / Output | The user will input information to create an account | Creates a user account |
| Login | Input / Output | Takes an email and password and matches them to an account already in existence | Logs a user into their account |
| Dashboard Page | Input / Output | Displays the most common system commands necessary to operate OSS | Main user interface |
| Register Object | Input | Displays a form necessary for adding an object to the system inventory | Adds an item / crate to the inventory system |
| Edit Object | Input / Output | Displays object data in an editable form | Item / crate will have their data updated |
| Delete Object | Output | Presents user with a confirmation box to verify object deletion | Item / crate will be removed from the system |
| Add Project | Input | Displays a form necessary for adding a project to the system inventory | A project will be added to the system |
| Edit Project | Input / Output | Displays project data in an editable form | A project will have their data updated |
| Delete Project | Output | Presents user with a confirmation box to verify project deletion | Project will be removed from the system |
| Item Description | Output | The system will show an item’s data fields | User will see item information |
| Crate Description | Output | The system will show the crate’s data fields | User will see crate information. |
| Project Description | Output | The system will show the project’s data fields | User will see project description and the items necessary to perform it |
| Check Out Objects | Input / Output | Displays a check out request form | The user will check out items/crates. |
| Check In Objects | Input / Output | The system will check items/crates back in to the system | The user will check in items/crates. |
| Item Request | Input / Output | A user will enter information about the item they are requesting to check out | Sends an item request form to the Admin |
| Request Management | Input / Output | Displays a user’s item request form, the admin will then approve/deny an item request | The item request will either be approved or denied. |
| Registration Management | Input / Output | Displays new user registrations that are currently pending | A new user’s registration will either be approved or denied. |
| Search | Input / Output | User will input information for the item to be searched | Web page will display search results. |
| RFID Reader (antenna integrated) | Input | Reader will detect RFID tag information. | Tag information will be sent to the server |
| Server | Input / Output | Will store software system and connect to the host computer via the internet | Will store / retrieve all inventory information and system software |
| Database | Input / Output | The database stores all OSS related data and can be accessed or changed by the system. | The system shall be able to modify the database through queries. |

Table 2‑1 Table of inputs and outputs

## 2.3 Product Interfaces

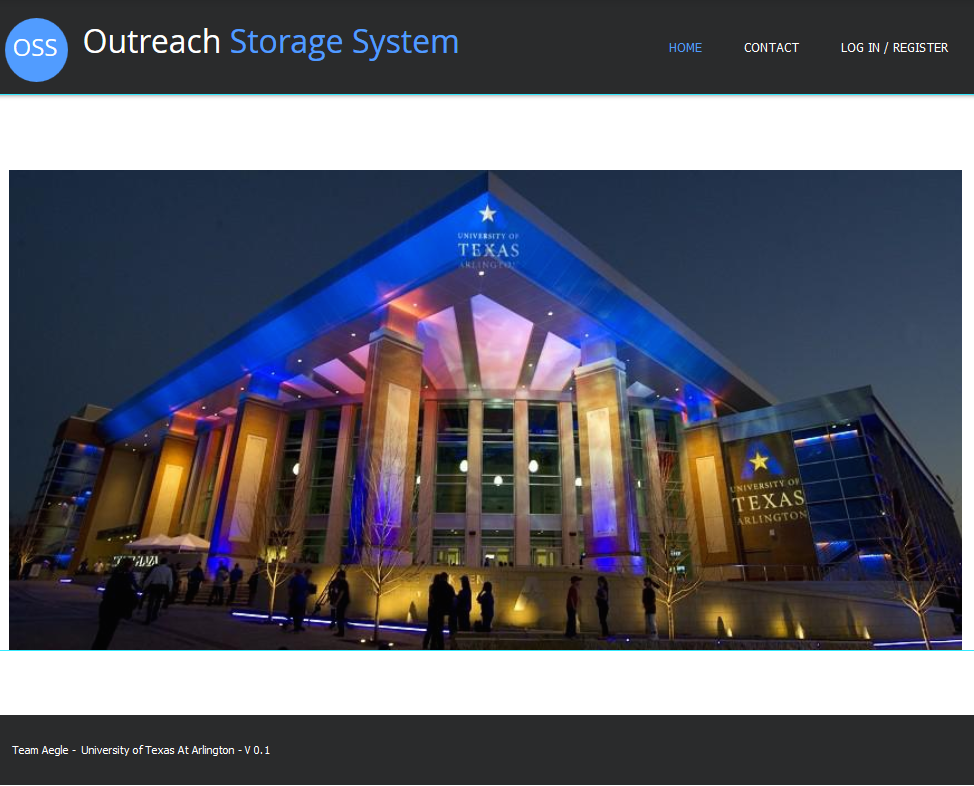


Figure ‑ Mockup prototype for Home page

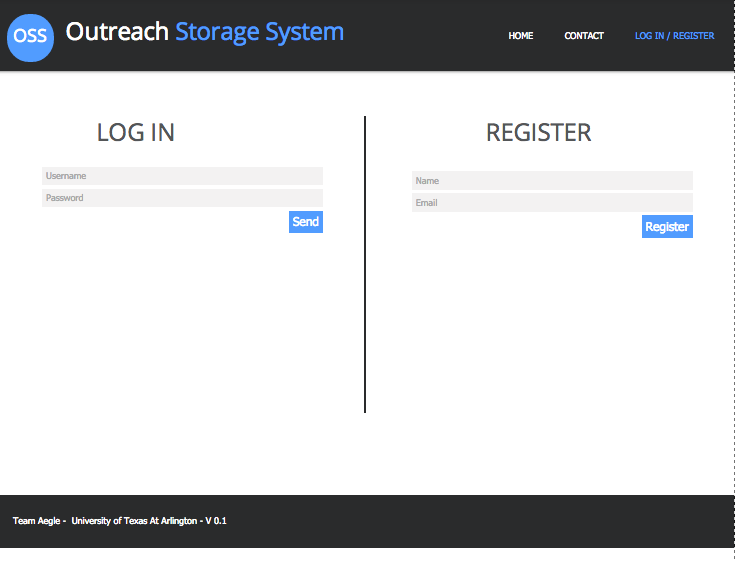


Figure ‑ Mockup prototype for Login / Register page

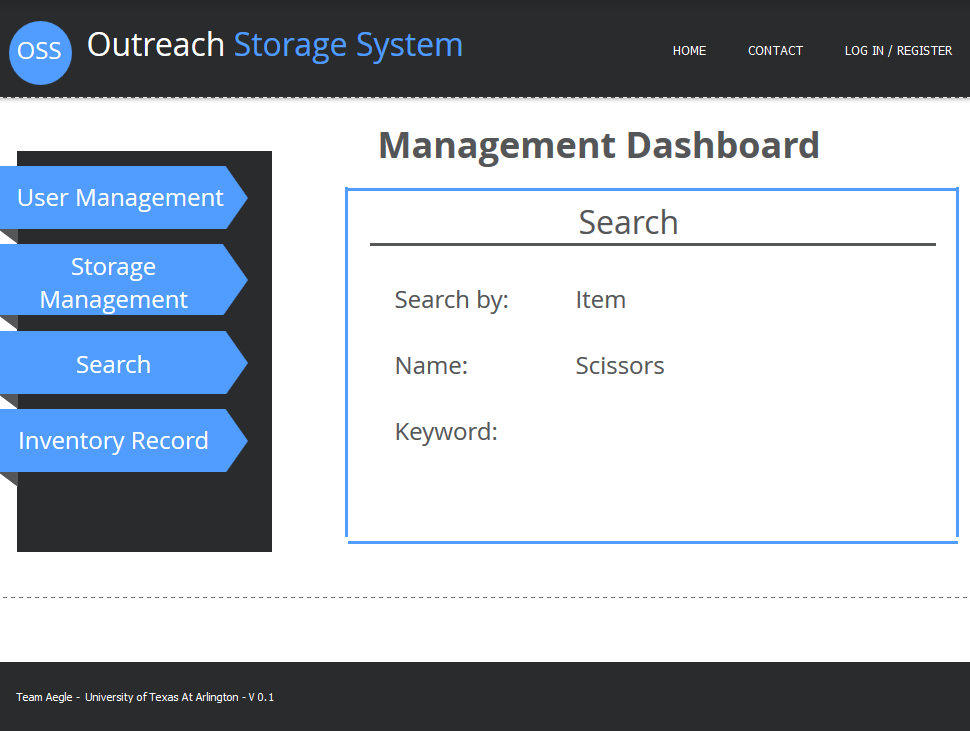


Figure ‑ Mockup prototype for Search page

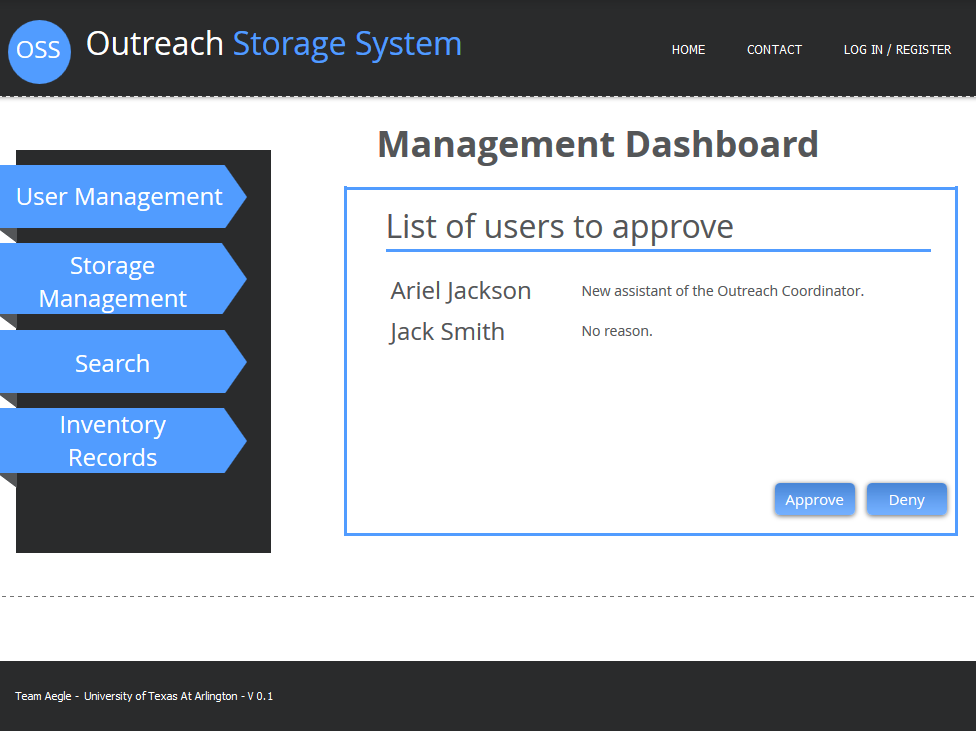


Figure ‑ Mockup prototype for User Management page

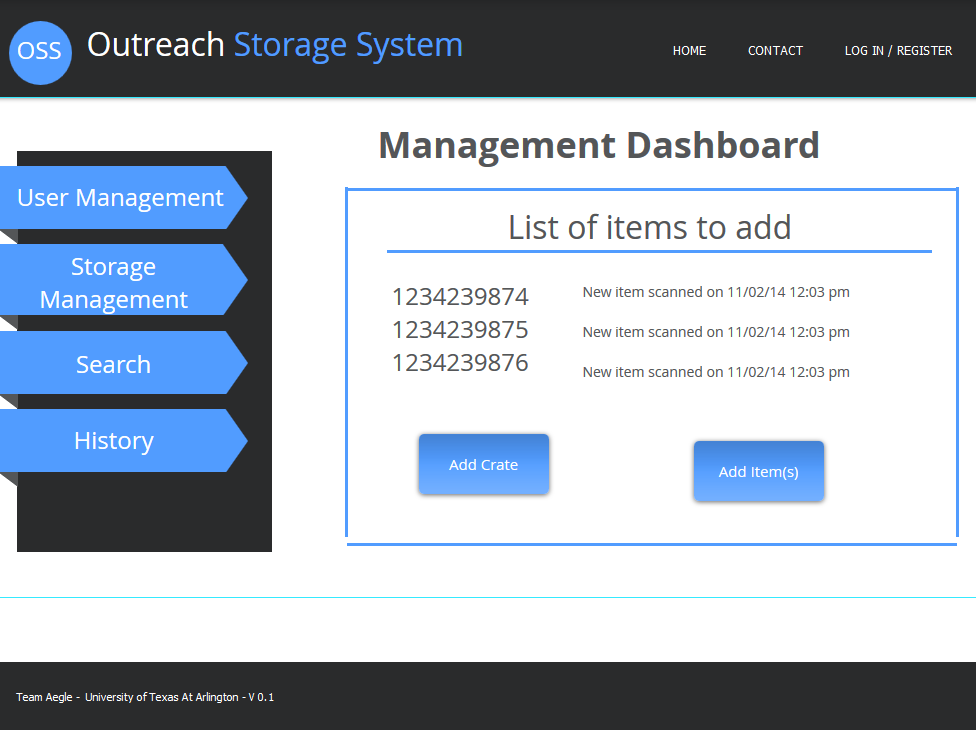


Figure ‑ Mockup prototype for Store Management page

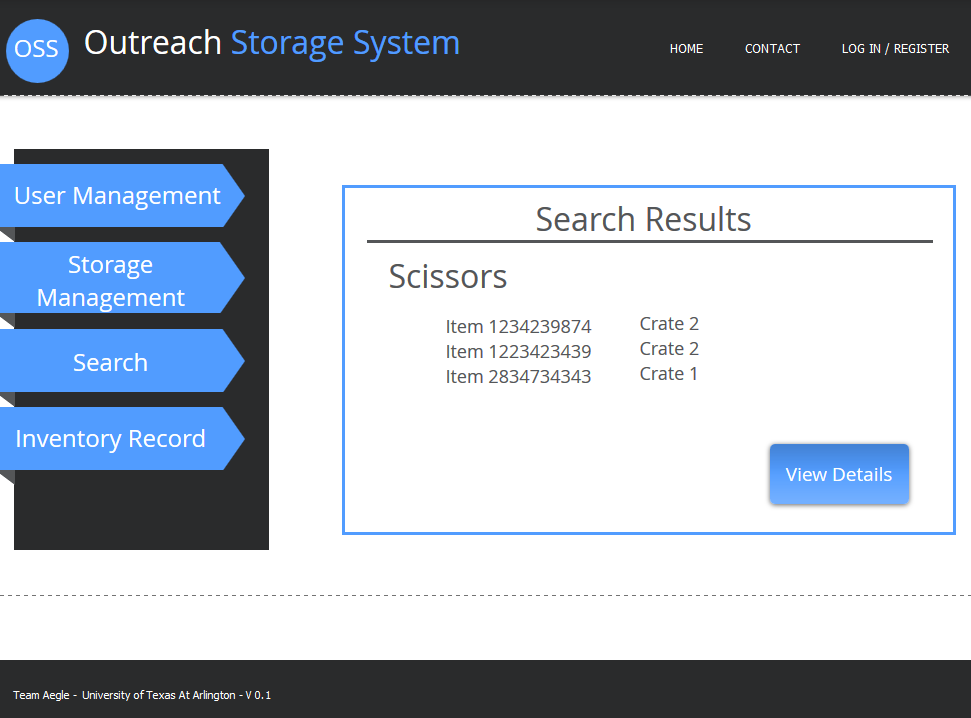


Figure ‑ Mockup prototype for Search Results page

# Customer Requirements

The following section covers the requirements that the system shall provide to the user. The Outreach Storage System will be a smart inventory system capable of providing updates of the items in inventory with description of item, location, and quantity of the item.

**Definitions:**

* **Crate:** Defined as a plastic storage bin, which contains one or more items and can be stored in a shelf located in the storage room. Crates can be checked out from the storage room. Crates can have either one crate category or one project assigned to it. The crates have the following data fields in the system:
  + **Unique crate ID number.**
  + **List of items inside of the crate.**
  + **Crate location inside storage room.**
  + **Crate status:** A crate has two statuses.
    - **Inventory:** The crate is available inside the storage room.
    - **Out of Storage Room at User’s Request:** The crate has been checked out of the storage room by a user. A request form must be submitted by the user and approved by an administrator before the crate can be set to this status. Request form defined later in this section.
  + **Crate Category or Project:** Optional field.
* **Item:** Defined as an object (i.e. Scissors) or collection of objects (i.e. 5 scissors in a bag, a box of Popsicle sticks, etc.) stored inside a crate. Items can be checked out from the storage room. Every item must have an RFID tag. Items can have either one crate category or one project assigned to it. Items can have up three alternate names, but only one main name. The items have the following data fields in the system:
  + **Unique item ID number.**
  + **Item name/s:** Some items may have multiple names.
  + **Item quantity.**
  + **Unique crate ID number:** Crate where the item is stored.
  + **Item status:** An item has two statuses.
    - **Inventory:** The item is available inside the storage room.
    - **Out of Storage Room at User’s Request**: The item has been checked out of the storage room by a user. A request form must be submitted by the user and approved by an administrator before the item can be set to this status. Request form defined later in this section.
  + **Project:** Optional field.
* **Request Form:** Defined as a form that is attached to an item or crate’s system description when it goes from Inventory to Out of Storage Room at User’s Request. The form will contain the following data fields:
  + **Administrator:** administrator who authorized the item or crate request.
  + **Registered user:** Registered userwho is submitting the request of the item or crate. Registered user could request items or crates on behalf of requesters.
  + **Requester:** A non-registered user on whose behalf a registered user is requesting an item for.
  + **Checkout Date:** The date on which an item or crate will be checked out of the storage room.
  + **Date of Return:** Planned date of item or crate return.
  + **Organization:** Name of the organization that the person who is requesting the item or crate belongs to.
  + **Reason:** The reason why the item or crate is being requested from the storage room.
* **Project:** A collection composed of one or more items required to perform a certain activity (i.e. “Flintstones”). Each project will maintain a list and quantity of the items required for the project.
* **Crate category:** Crates can be organized with a crate category in order to store items of similar characteristics together (i.e. Crafts, Electronics, Tools, etc.).

**Users:**

* **Administrator:** The role of the administrator will be to manage the items and crates in the storage room. The administrator shall have the capability of adding and removing items, crates, and projects to/from the storage room. Administrators shall also be able to edit the data fields of items, crates, and projects stored in the system. In addition, the administrator also has the capability to deny or accept any request to check out items or crates to a registered user. The administrator will be able to view the location of the items and crates in the storage room. The system shall support multiple administrators, but only one Main administrator (Dr.Tiernan) can add other administrators to the system.
* **Registered User:** The role of the registered user is to request or return items and crates to the storage room. They will also be able to view the current items or crates located in the storage room. The registered user shall only be able to retrieve items from the storage room after the administrator has accepted the item request.
* **Non-registered user:** A person wanting to register into the system.
* **Requesters:** Any person that requests an item or crate from the storage room. A requestor does not have to be a registered user.
* **Users:** administrators, registered users, and Non-registered users.

**Assumptions:**

* The registered user or administrator must place items returning to the storage in their respective crate. If the item belongs to a crate with a crate category and the crate is empty, the item can be placed in a different crate. If there are remaining items of the same kind in the crate then the item must be placed in the crate that contains the remaining items.
* If an item is placed into a different crate it must be updated in the system.
* Each item shall have a unique RDIF tag.
* The crate’s material should not interfere with the radio frequency of the RFID reader.
* All items and crates must be initialized in the system.
* The client shall provide host computer.

## 3.1 Keep track items and crates by the system.

### 3.1.1 Description: The system shall be able to keep track of the items’ status (in stock, or out of stock) and keep track of crates’ status.

### 3.1.2 Source: Team Aegle.

### 3.1.3 Constraints: An item must have an RFID tags. The system must have a valid connection between the RFID Reader and the server.

### 3.1.4 Standard: Federal Communication Commission (FCC).

### 3.1.5 Priority: 1 – Critical.

## 3.2 System Description of Items

### 3.2.1 Description: The system shall be able to provide a description of the item to the administrators and registered users. The description shall provide the item data fields stated previously on definitions section.

### 3.2.2 Source: Sponsor, Dr. Tiernan.

### 3.2.3 Constraints: administrator or registered user must be logged in. Valid Internet connection to access website application.

### 3.2.4 Standard: None.

### 3.2.5 Priority: 1 – Critical.

## 3.3 System Description of Crates

### 3.3.1 Description: The System shall be able to provide a description of the crate to the administrators and registered users. The description will provide crate data fields stated previously on definitions section.

### 3.3.2 Source: Sponsor, Dr. Tiernan.

### 3.3.3 Constraints: administrator or registered user must be logged in. Valid Internet connection to access website application.

### 3.3.4 Standard: None.

### 3.3.5 Priority: 1 – Critical.

## 3.4 Search Function for Items and Crates

### 3.4.1 Description: The system shall be able to search the database for items and crates by the multiple data fields of the items and crates describe in the definitions section and by general word search.

### 3.4.2 Source: Sponsor, Dr. Tiernan.

### 3.4.3 Constraints: administrator or registered user must be logged in. Valid Internet connection to access website application.

### 3.4.4 Standard: None.

### 3.4.5 Priority: 1 – Critical.

## 3.5 Locating Item Inside a Crate

### 3.5.1 Description: The system shall be able to locate in which crate an item is located.

### 3.5.2 Source: Sponsor, Dr. Tiernan.

### 3.5.3 Constraints: Item must have an RFID tag. There has to be a valid connection between the RFID Reader and the RFID Middleware Server and for the server.

### 3.5.4 Standards: Federal Communication Commission (FCC).

### 3.5.5 Priority: 1 – Critical.

## 3.6 Item Management by the administrators

### 3.6.1 Description: The administrators shall be able to add, remove and delete an items in the System.

### 3.6.2 Source: Sponsor, Dr. Tiernan.

### 3.6.3 Constraints: Valid Internet connection to access website application. administrators must be logged in.

### 3.6.4 Standard: None.

### 3.6.5 Priority: 1 – Critical.

## 3.7 Crate Management by the administrators

### 3.7.1 Description: The administrators shall be able to register a new crate into the system, as well as delete and edit crates already existing in the system.

### 3.7.2 Source: Team Aegle.

### 3.7.3 Constraints: Valid Internet connection to access website application. Administrators must be logged in.

### 3.7.4 Standard: None.

### 3.875 Priority: 3 – Moderate.

## 3.8 Project Management by administrators

### 3.8.1 Description: The administrators shall be able to create, edit and delete projects in the system.

### 3.8.2 Source: Sponsor, Dr. Tiernan.

### 3.8.3 Constraints: Valid Internet connection to access website application. administrator must be logged in.

### 3.8.4 Standard: None.

### 3.8.5 Priority: 2 – High.

## 3.9 System Interaction by administrators

### 3.9.1 Description: The administrator shall be able to look at the items and crates in the inventory, look at the different projects, and shall be able to perform any functionality specified in other requirements. The administrator shall be able to approve or deny a registered user’s request to check out an item(s) or crate(s) from the inventory. Also, the administrator shall be able to check out an item(s) or crate(s) from inventory and approve or deny a user registration request.

### 3.9.2 Source: Team Aegle.

### 3.9.3 Constraints: administrators must be logged in. Valid Internet connection to access website application.

### 3.9.4 Standard: None.

### 3.9.5 Priority: 2 – High.

## 3.10 System Interaction by Registered Users

### 3.10.1 Description: Registered users shall be able to look at the items in inventory, and shall be able to perform any functionality specified in other requirements. Registered users shall be able to look at the different projects, and submit a request for permission to take an item(s) or crate(s) from inventory, from the administrator.

### 3.10.2 Source: Sponsor, Dr. Tiernan.

### 3.10.3 Constraints: Registered user must be logged in. Valid Internet connection to access website application.

### 3.10.4 Standard: None.

### 3.10.5 Priority: 1 – Critical

## 3.11 Registration/Login System

### 3.11.1 Description: The system shall have a registration and login system, which will be the only way to access the database and system functionalities. The registration requirements will be an email account, first name, last name, date of birth, phone number, organization, and password. The users shall be able to login into the system with their email account and password, after the administrator has approved their account.

### 3.11.2 Source: Team Aegle.

### 3.11.3 Constraints: Valid Internet connection.

### 3.11.4 Standard: None.

### 3.11.5 Priority: 1 – Critical.

## 3.12 Web-Based Accessible Application

### 3.12.1 Description: The system shall be implemented as a web-based application.

### 3.12.2 Source: Team Aegle.

### 3.12.3 Constraints: Google Chrome and Mozilla Firefox are the only browsers supported by the website application.

### 3.12.4 Standard: IETF Standards.

### 3.12.5 Priority: 1 – Critical.

# Packaging Requirements

This section specifies all of the packaging requirements that must be met by the Outreach Storage System. A user manual with all the software required will be provided in order to do self-installation of the RFID system.

## 4.1 Included Components

### 4.1.1 Description: The final product shall include the following components: Passive RFID tags for the items, RFID reader, and RFID middleware server.

### 4.1.2 Source: Team Aegle.

### 4.1.3 Constraints: None.

### 4.1.4 Standards: FCC Standards for RFID middleware software.

### 4.1.5 Priority: 1 – Critical.

## 4.2 Installation Manual

### 4.2.1 Description: The final product shall provide an installation manual that includes detailed instructions on how to install, set up, and use the system. The final product shall also include any required software.

### 4.2.2 Source: Team Aegle.

### 4.2.3 Constraints: None.

### 4.2.4 Standards: None.

### 4.2.5 Priority: 1 – Critical.

## 4.3 User Manual

### 4.3.1 Description: The final product shall provide a user manual that includes detailed information on system features and instructions on system functionalities.

### 4.3.2 Source: Team Aegle.

### 4.3.3 Constraints: None.

### 4.3.4 Standards: None.

### 4.3.5 Priority: 2 – High.

## 4.4 Power Supply

### 4.4.1 Description: The final product shall include a power supply for the RFID reader.

### 4.4.2 Source: Team Aegle.

### 4.4.3 Constraints: None.

### 4.4.4 Standards: None.

### 4.4.5 Priority: 1 – Critical.

## 4.5 Range of RFID Antenna

### 4.5.1 Description: The RFID reader antenna range shall be able to cover the door length.

### 4.5.2 Source: Team Aegle.

### 4.5.3 Constraints: The RFID reader / writer with the integrated antenna, must be placed in a static position within the doorframe.

### 4.5.4 Standards: FCC Standards.

### 4.5.5 Priority: 1 – Critical.

# Performance Requirements

The following section covers the performance requirements for the project. The primary concerns for performance in our project lies in the responsiveness of the interface and the usability of the product in conditions where the system is deprived of resource that it uses to operate. It is expected that the product will have a responsive and streamlined feel to it, which these requirements mandate.

## 5.1 Check-in/Check-out Latency

### 5.1.1 Description: The amount of time that it takes for the system to recognize that an item has entered or left the storage room should not exceed 10 seconds.

### 5.1.2 Source: Team Aegle.

### 5.1.3 Constraints: Network connectivity and speed

### 5.1.4 Standards: None

### 5.1.5 Priority: 2 – High.

## 5.2 Web Interface Response Latency

### 5.2.1 Description: The amount of time that it takes for the web application to return meaningful information to the user shall not exceed 10 seconds.

### 5.2.2 Source: Team Aegle.

### 5.2.3 Constraints: Network connectivity and speed

### 5.2.4 Standards: None.

### 5.2.5 Priority: 2 – High

# Safety Requirements

The following section outlines the requirements that protect the system’s end users from physical and monetary damages related to the OSS.

## 6.1 Electrical Hazard

### 6.1.1 Description: The system and its components present within the storage room shall not pose an electrical hazard to its users or the building it resides in.

### 6.1.2 Source: Team Aegle.

### 6.1.3 Constraints: None.

### 6.1.4 Standards: None.

### 6.1.5 Priority: 1 – Critical.

## 6.2 Signal Interference

### 6.2.1 Description: The RFID system shall not interfere with any critical radio frequency transmission.

### 6.2.2 Source: Team Aegle.

### 6.2.3 Constraints: FCC signal interference guidelines

### 6.2.4 Standards: None

### 6.2.5 Priority: 3 – Moderate.

# Maintenance and Support Requirements

The following section specifies the features and help that the end user will be provided with upon product completion. It is not expected that the developers of this product will provide any personal support after the project’s completion, but it is expected that the end users will have the necessary documentation and instructions to fix a simple problem, or when a third party needs to be involved

## User Manual

### 7.1.1 Description: The final product shall come with a user manual describing in detail how to set up the system and use its various features.

### 7.1.2 Source: Team Aegle.

### 7.1.3 Constraints: None.

### 7.1.4 Standards: None.

### 7.1.5 Priority: 3 – Moderate.

## 7.2 Troubleshooting Guide

### 7.2.1 Description: The final product shall include a troubleshooting guide to help solve general problems the user may have, and to assist them in determining whether the problem needs to be solved by a third party.

### 7.2.2 Source: Team Aegle.

### 7.2.3 Constraints: None.

### 7.2.4 Standards: None.

### 7.2.5 Priority: 3 – Moderate.

## 7.3 Source Code Availability & Documentation

### 7.3.1 Description: The final product shall include all the source code and documentation used to design and implement the system. The source code will be well structured and commented so as to allow for future modularity and support.

### 7.3.2 Source: Team Aegle.

### 7.3.3 Constraints: None.

### 7.3.4 Standards: None.

### 7.3.5 Priority: 3 – Moderate.

# Other Requirements

The following section describes all the requirements that did not fall directly under any of the previous sections of this document will be found here.

## 8.1 Security

### 8.1.1 Description: The system shall ensure the privacy and security of personal information being stored and transmitted by users of the system through the use of secure connectivity and secure programming techniques.

### 8.1.2 Source: Team Aegle.

### 8.1.3 Constraints: None

### 8.1.4 Standards: None.

### 8.1.5 Priority: 3 – Moderate.

## 8.2 Connectivity Tolerance

### 8.2.1 Description: The system shall be tolerant of a loss of communication with the web server. As many transactions that a user’s device can buffer, will be processed and stored when the server comes back up, even during system down time.

### 8.2.2 Source: Team Aegle.

### 8.2.3 Constraints: End User device’s storage buffer

### 8.2.4 Standards: None.

### 8.2.5 Priority: 2 – High.

# Acceptance Requirements

The following section describes the different procedures that validate the correct functionality of the OSS to the customer. Every acceptance requirement should be valid in order for the product to be approved by Aegle and our customer. As the project develops, the criteria will be redefined to accommodate any upcoming modifications.

## 9.1 Verify User Account Registration

### 9.1.1 Requirements Addressed: Requirements 3.10, 3.11, 3.12, and 3.13. System must be able to verify that a new user can be registered and be able to login.

### 9.1.2 Verification Procedure: The Home page in the web application will have a “Register” button that allows the user to send a registration request to the administrator, who will either accept or deny the user to be added to the system. After the administrator logs into the system and he will navigate to the “User Management” section, where a list of user requests will be displayed. When a user is selected, the administrator will click on the “Register” button and the user should be listed in the approved users list. Lastly, the new user will successfully log in to the system using email and password.

## 9.2 Verify Tracking an Item Functionality

### 9.2.1 Requirements Addressed: Requirements 3.1, 3.2, 3.4, 3.5, 3.10, 3.11, 3.12, and 3.13. System must be able track an item in the storeroom and locate which crate it is stored in.

### 9.2.2 Verification Procedure: This requirement will be verified by searching on a registered item that will accurately display the crate where the item is contained. First, an authorized user will login to the system and click on the “Search” menu in the “Dashboard Management” page. The user will be redirected to the “Search” page, where search criteria will be specified. After clicking on the “Search” button, the search results will be displayed with a list of items that matched the search criteria. Clicking on the button “See details” will show a more detailed description of the selected item.

## 9.3 Verify Tracking a Crate Functionality

### 9.3.1 Requirements Addressed: Requirements 3.1, 3.3, 3.4, 3.10, 3.11, 3.12, and 3.13. System must be able to track a crate inside the storeroom, including its contents and its location.

### 9.3.2 Verification Procedure: This requirement will be verified by searching on a registered crate that will accurately display the shelf where the crate is placed. First, an authorized user will login to the system and click on the “Search” link in the navigation bar. The user will be redirected to the “Search” page, where search criteria will be specified. After clicking on the “Search” button, the crate will display accurate information of where the crate is located, providing a description and the shelf where the crate is located.

## 9.4 Verify Search Functionality

### 9.4.1 Requirements Addressed: Requirements 3.2, 3.4, 3.5, 3.6, 3.10, 3.11, 3.12, and 3.13. System must be able to search for an item in the storeroom, including which crate it is located in.

### 9.4.2 Verification Procedure: The “Search” button in the web application will search for an item in the storage room. First, the administrator or registered user will log in to their respective accounts. Next, the user will click on the “Search” link, located in the navigation bar. The user will be redirected to the “Search” page where different search criteria will be selected and correctly filled out. Last, the user will click on the “Search” button and the system will display the item/crate description along with the corresponding location.

## 9.5 Verify Item Management Functionality

### 9.5.1 Requirements Addressed: Requirements 3.4, 3.7, 3.10, 3.11, 3.12, and 3.13. System must be able to manage items stored in the storeroom such as adding and removing items.

### 9.5.2 Verification Procedure: An authorized user must register an item in to the system along with the corresponding crate that will contain it. Once the item is successfully registered in the system, a search will be performed to make sure the item is correctly displayed by the system. Next, the authorized user will select the item in the search results and click on the “Edit” button. The user will be redirected to the “Edit Item” section, where he will be able to edit the item name or description. After editing, the user clicks on the “Save” button. The item description or name should display the new information. Lastly, the user will search for the item again. On the search results, the item will be selected and click on the “Remove” button. A confirmation alert will be displayed with two buttons: a “Cancel” button that will cancel the deletion of the item and a “Remove” button that will permanently remove the item from the system. If a new search is performed on the item, the item should no longer be displayed in the system.

## 9.6 Verify Crate Management Functionality

### 9.6.1 Requirements Addressed: Requirements 3.4, 3.8, 3.10, 3.11, 3.12, and 3.13. System must be able to manage crates in the storeroom including adding and removing crates.

### 9.6.2 Verification Procedure: An authorized user must register a crate in to the system along with the corresponding shelf where the crate will reside. Once the crate is successfully registered in the system, a search will be performed to make sure the crate is successfully registered in the system. Next, the authorized user will select the crate in the search results and click on the “Edit” button. The user will be redirected to the “Edit Crate” section, where he will be able to edit the crate name or description. After editing, the user clicks on the “Save” button. The crate description or name should display the new information. Lastly, the user will search for the crate again. On the search results, the crate will be selected and click on the “Remove” button. A confirmation alert will be displayed with two buttons: a “Cancel” button that will cancel the deletion of the crate and a “Remove” button that will permanently remove the crate from the system. If a new search is performed in the crate, the crate should no longer be displayed in the system.

## 9.7 Verify Project Management Functionality

### 9.7.1 Requirements Addressed: Requirements 3.9, 3.10, 3.11, and 3.13. System must be able to manage projects such as adding/removing and editing projects.

### 9.7.2 Verification Procedure: An authorized user must login to the system and navigate to the “Projects” page. Once the user navigated to the page, he will click on the “Add New Project” button and fill out the Project Form. Once the form is correctly filled out, the user will click on the “Save” button. The Project will display in the Projects list. User will select the recently created Project and click on the “Edit Project” button. The system will display the Project Form where the user will edit information and click on the “Save” button. The Project changes will be displayed correctly. Finally, the user will select the same project and click on the “Remove” button. A confirmation alert will be displayed with two buttons: a “Cancel” button that will cancel the deletion of the crate and a “Remove” button that will permanently remove the Project from the system. The Project will no longer be visible in the Projects List.

# Use Cases

The following section will cover how the users interact with the OSS via the web-based application. The use cases will assume that the system has been installed, setup and ready for use.

Definitions: TUCBW: This Use Case Begins With. TUCBW: This Use Case Ends With.

## 10.1 Registering into the System

### 10.1.1 Scenario: A non-registered user clicks on the “Log In / Register” button while in the web-based application. This action will redirect the person to the log in/ registration webpage. In this webpage, the non-registered user must fill the information and click complete registration in order to complete the registration process. The following information is needed by the non-registered user in order to register: email, first name, last name, date of birth, phone number, organization, and password.

### 10.1.2 Actor(s): Non-registered users.

### 10.1.3 TUCBW: The non-registered user clicks on the register button.

### 10.1.4 TUCEW: The non-registered user is notified that the registration was successful.

## 10.2 Logging into the System

### 10.2.1 Scenario: Either an administrator or registered user is in the home page of the web-based application; it fills the login box information with their email and password, and clicks on Log In or presses enter to complete the login process.

### 10.2.2 Actor(s): administrators or registered users.

### 10.2.3 TUCBW: The administrator or registered user opens the web-based application.

### 10.2.4 TUCEW: The administrator or registered user is logged in to the system.

## 10.3 Adding Item, Crate, or Project to the System

### 10.3.1 Scenario: The administrator clicks on the Add item link, Add crate link, or Add Project link while in the Inventory webpage. This will prompt the administrator to fill out a form for the new item, crate, or Project to be added into the System. The administrator must fill out the form and click Complete. After clicking complete the administrator will redirect to the List of items, List of crates, or List of Projects with the newly added item/crate/Project highlighted and showing the item/crate/Project description below it.

### 10.3.2 Actor(s): administrators

### 10.3.3 TUCBW: The administrator clicks on Add item link, Add crate Link, or Add Project Link.

### 10.3.4 TUCEW: The administrator is able to see the item, crate, or Project and the description in the item List, crate List, or Project List.

## 10.4 Editing Item, Crate, or Project in the System

### 10.4.1 Scenario: The administrator clicks on the Edit link either for an item in the items List, for a crate in the crates List, or Project in the Projects List. After this action the administrator is redirected to a website where it can make any changes to the item\crate\Project exceptions are to change the ID# of an item and crate. After making the changes, the administrator must click on complete in order to finish editing. After that the administrator is redirected to List of items, List of crates or List of Project.

### 10.4.2 Actor(s): administrators.

### 10.4.3 TUCBW: The administrator clicks on the Edit link for an item, crate, or Project.

### 10.4.4 TUCEW: The administrator is redirected to the List of items, crates, or Projects.

## 10.5 Deleting Item, Crate, or Project from the System

### 10.5.1 Scenario: The administrator clicks on the Delete link either for an item in the items List, for a crate in the crates List, or Project in the Projects List. After this action the administrator will be asked to a final confirmation. After clicking either Yes or No, the administrator is redirected to the items List, crates List, or Projects List.

### 10.5.2 Actor(s): administrators.

### 10.5.3 TUCBW: The administrator clicks on the Delete link for an item, crate, or Project.

### 10.5.4 TUCEW: The administrator is redirected to the list of items, crates, or projects webpage.

## 10.6 Searching for Items and Crates

### 10.6.1 Scenario: The administrator or registered users types in a word in the search box and presses Enter or clicks on the Search button. Another way to search is by clicking advanced search, filling in the search criteria, and clicking on Search. Once the administrator or registered user clicks on Search, they are directed to the search results.

### 10.6.2 Actor(s): The administrator or registered users.

### 10.6.3 TUCBW: The administrator or registered user does a search by word or an advanced search.

### 10.6.4 TUCEW: The administrator or registered users are shown the search results.

## 10.7 User Registration Confirmation by administrator

### 10.7.1 Scenario: The administrator will receive notifications about User Registration that they will have to approve or deny. The administrator can review the registration before denying or approving it. The administrator will click on the notification, which will take the administrator to a webpage where it can view the User Registration information and then it can click on Approve or Deny.

### 10.7.2 Actor(s): administrator.

### 10.7.3 TUCBW: administrator clicks on Registration Notification.

### 10.7.4 TUCEW: administrator is notified that the Registration was successfully approved or denied.

## 10.8 Item/Crate Check-out Approval by administrator

### 10.8.1 Scenario: The administrator will receive notifications about registered users request to checkout item(s)\crate(s) from the storage room. The administrator will have to approve or deny the request. The administrator will click on the checkout request notification to see the request form for this request.

### 10.8.2 Actor(s): Administrator.

### 10.8.3 TUCBW: The administrator clicks on the checkout request notification.

### 10.8.4 TUCEW: The administrator is notified that it successfully approved or denied the checkout request.

## 10.9 Item/Crate Checkout Request by Registered User

### 10.9.1 Scenario: The registered users can create item(s)\crate(s) Checkout Requests to the administrator by Selecting the item(s)\crate(s) and then Clicking on checkout. The registered users must fill out the request form.

### 10.9.2 Actor(s): Registered users.

### 10.9.3 TUCBW: The registered user click on Checkout after selecting the item(s)\crate(s) to be checked out.

### 10.9.4 TUCEW: The Request is sent to the administrator.

## 10.10 Item\Crate Checkout by administrator

### 10.10.1 Scenario: The administrator selects the item(s)\crate(s) that they want to check out and then they click on Check out, the administrator must fill out a Request Form after Clicking on Check-Out, and then Click Confirm to finish the Check-Out.

### 10.10.2 Actor(s): administrator.

### 10.10.3 TUCBW: The administrator selects the item(s)\crate(s) and clicks on Check out.

### 10.10.4 TUCEW: administrator sees a confirmation message.

## 10.11 Item\Crate Check-in by administrators and Registered Users

### 10.11.1 Scenario: The administrator or registered user returning an item must place the item in its respective crate, and update the quantity of the item if necessary in the web application. Also if the administrator or registered user places the item in a different crate, they must update the item location change in the web application. Returning a crate will be just as returning an item, the quantity of the items’ inside the crate must be updated if necessary and if the location of the crate changed, the crate location must be updated in the web application.

### 10.11.2 Actor(s): administrators and registered users.

### 10.11.3 TUCBW: The administrator or registered user brings an item or crate to the Storage Room.

### 10.11.4 TUCEW: The administrator or registered user gets a notification that the item or crate was successfully returned.

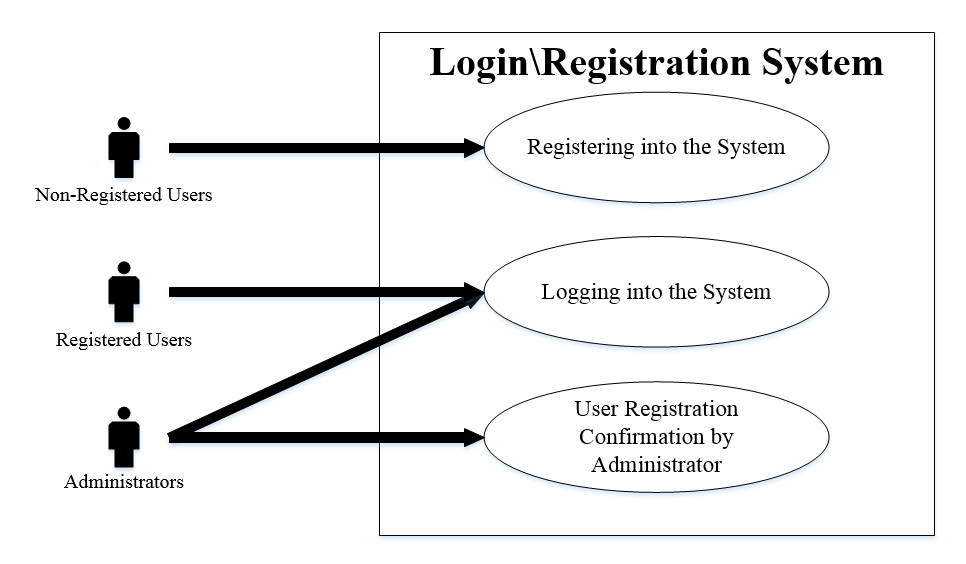


Figure ‑ Login and Registration System

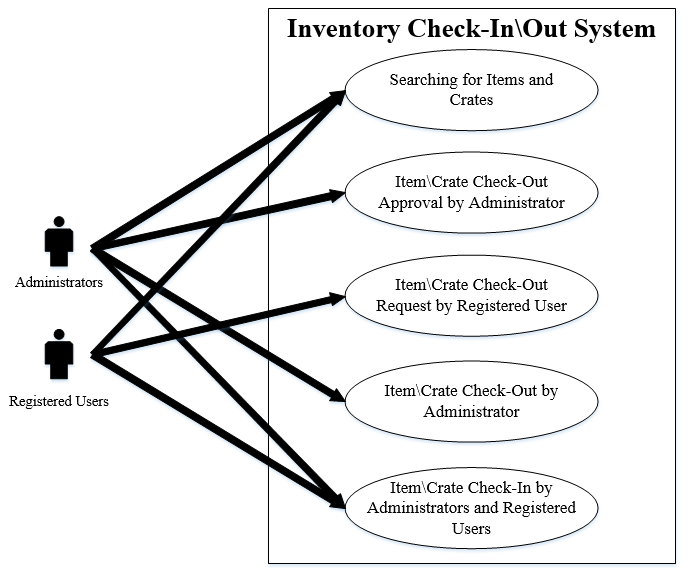


Figure ‑ Inventory check in /out system

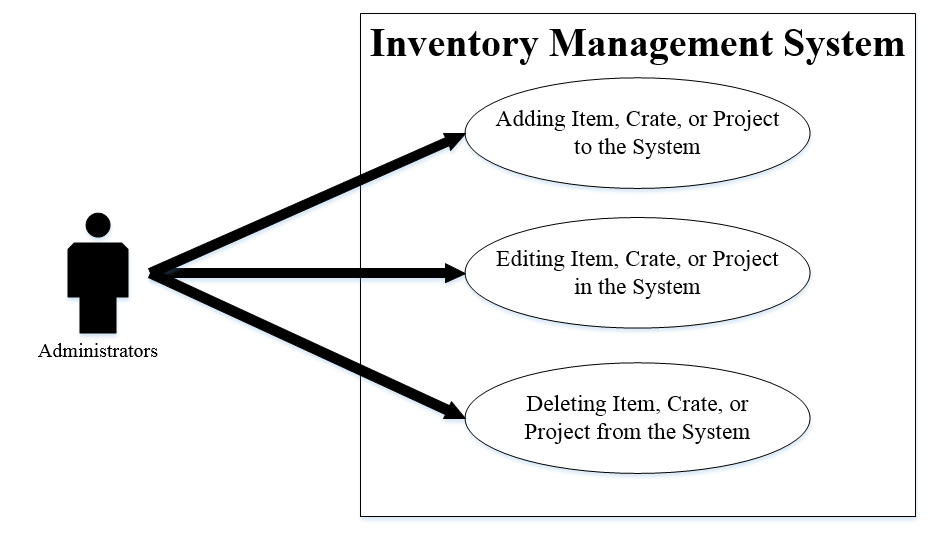


Figure ‑ Inventory Management

# Feasibility Assessment

The following section provides a Feasibility Assessment of the Outreach Storage System, according to Aegle’s estimation experience acquired in previous projects. The assessment is based in six different sections: Scope Analysis, Research, Technical Analysis, Cost Analysis, Resource Analysis and Schedule Analysis.

## 11.1 Scope Analysis

The Scope of the Outreach Storage System, based on the critical requirements provided by the sponsor, is reasonable given the time frame, budget and resources available. The prototype system will be composed of an RFID reader that has an integrated antenna, RFID tags, crates, items, and a server (that will be provided by the user). The software will be installed in the server (team Aegle will use their personal server for testing). The most critical requirement is being able to determine the count of the items in the storage room by successfully sending data from the RFID reader through a network and registering correctly in the server. The web application has to be able to display all the items or crates and its location. Requirements are ordered by priority, from most critical to low priority, to ensure that critical requirements are addressed in the early stages of development in case any issue arises.

## 11.2 Research

Our development team has done intensive RFID technology research in order to understand how data is passed through radio frequency waves and how we can this data can be transmitted through a network. Defining the type of RFID reader/writer that we will implement is very important to our project, since it has a high cost associated; therefore an informed decision needs to be made. We have identified different types of RFID readers that can be implemented in the storage room; they could be stationary or handheld device. Further research on this technology, is indicating a lean towards using a stationary RFID reader/writer. This will facilitate the readability of multiple RFID tags at the same time without reader collusion problems.

The University of Texas at Arlington has an RFID laboratory managed by Dr. Jones and Ph.D. students who contribute with the research activities. The team was exposed in a meeting to this technology to have a better understanding of how it works and also to help create a better design of the hardware piece of the system.

As a result of the research performed by the team, the Outreach Storage System will need a stationary RFID reader/writer with an integrated antenna, RFID passive tags, a valid network connection, and a host server where the software will be installed. These pieces will be critical for the development of the project.

## 11.3 Technical Analysis

The type of RFID reader that we will implement, will dictate the manner in which the items/crates will be registered into the application. Therefore it is one of the most important decisions that the team has to make since most of the critical requirements depend on an RFID reader being able to read/write and send data through a network transferred in packages. According to the research that the team we have done, we think that a stationary RFID reader is more viable for the type of operations that are going to be performed in the storage room.

The OSS application will be responsible for receiving and parsing any data that the RFID reader transmits trough the network in standard format such JSON, Binary Data, XML, etc.

The items, crates and shelves will be assigned a unique RFID passive tag. Passive tags have an infinite lifespan due to the fact that an RFID reader powers its internal battery when it is in close range.

The system will integrate a restful API, which will be responsible for storing the data captured by the RFID reader/writer to a relational database. This API will also communicate with the web client application, providing the customer with a visible state of the inventory. The server that the team will be use for testing is going to run on Linux.

The web client application will also allow different types of users to perform a variety of tasks, depending on the user’s privileges. The web client application will be able to modify the state of the database if it’s operated through an authorized system.

## 11.4 Cost Analysis

According to the cost research that our team has done, it has been determined that the project can be successfully developed within the eight hundred dollars budget that the team has been allocated. The project is composed of both hardware and software. The hardware part of the project includes the RFID reader/writer with an integrated antenna, RFID passive tags, items and crates that will be used to test the project functionality. The software part of the project needs a server to host the restful API and the Web Client Application.

The following table shows relative prices of the components required for the development of the Outreach Storage System.

|  |  |
| --- | --- |
| **Parts** | **Cost** |
| RFID Reader/Writer | $445.00 |
| RFID Tags (30) | $20.00 |
| Miscellaneous items | $30.00 |
| Plastic crates (4) | $40.00 |
| **Total Cost** | $535.00 |

Table 11‑1: Preliminary Cost Analysis

## 11.5 Resource Analysis

Team Aegle consists of two software engineering students, one computer-engineering student and one computer science student.

The OSS is composed of two main branches, hardware and software. The hardware branch of the project consists on reading RFID tags, and correctly configuring the RFID reader so that it can transmit correctly data through a network. Our computer-engineering student, who has more experience in hardware technology, will lead this work. The computer science student has been exposed to some hardware; therefore he will be focused in both hardware and software since this part corresponds to around a 30% of the entire application.

As far as the software project, the two software engineering students will be mainly focused on developing the entire application, dividing the work in two, which will be the web client interface and the server side application. Finally, the computer science student will be the lead on the database implementation, due to his experience with relational databases in previous projects.

The fact that each team member has a role and a part of the application assigned doesn’t mean they are restricted to only work on that part of the application. When it comes to implementing new technologies and doing research, the whole team will be involved to make sure we all share the same knowledge and we are all in the same terms.

All our team members have experience writing software and doing architectural and database design, we feel very strong and comfortable in this part of the project, since architectural design and coding was one of our strengths and our project is heavily based in software.

As far as our weaknesses, only one of the team members has knowledge working with hardware components, so having only one team member experienced in this could potentially slow us down and delay our schedule. However, we will be on top of our risk management plan to make sure we stay on track and we are able to identify any potential issues related to hardware to attack them as soon as possible before they cause catastrophic damage to our project.

## 11.6 Schedule Analysis

The following section will talk about three different techniques that the team used to estimate how complex the project is and the time it would take to develop it, according to different factors that influence the design and implementation of the Outreach Storage System.

The first method of estimation that we implemented is the Jones First Order. The purpose of this method is to identify how big or complex the project is, in order for us to determine if we are able to successfully complete the project on the time frame available.

First, we identified all the inputs, outputs, inquiries, logical internal files and external interfaces. After that, we classify them as low complexity, medium complexity or high complexity, as demonstrated on the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Type** | **Low**  **Complexity** | **Medium Complexity** | **High**  **Complexity** |
| Number of inputs | 12 x 3 | 9 x 4 | 0 x 6 |
| Number of outputs | 4 x 4 | 3 x 5 | 0 x 7 |
| Inquiries | 0 x 3 | 1 x 4 | 0 x 6 |
| Logical internal files | 5 x 7 | 3 x 10 | 0 x 15 |
| External interface files | 0 x 5 | 0 x 7 | 1 x 10 |
| Unadjusted function-point total |  |  | 182 |

**Table 11‑2**: Function Point Table Breakdown

The previous table demonstrates the number of different functions that team Aegle identified as being part of the Outreach Storage System, with the intention of producing the Unadjusted functional point total, which turned out to be one hundred and eighty two.

After calculating the unadjusted functional point total, we assigned values to the influence multipliers, with the purpose of getting a more accurate estimation.

|  |  |
| --- | --- |
| **Adjustment factor** | **Degree of Influence (0-5)** |
| Data Communication | 5 |
| Distributed Data Processing | 0 |
| Performance | 1 |
| Heavily Used Configuration | 0 |
| Transaction Rate | 2 |
| On-line Data Entry | 4 |
| End-User Efficiency | 3 |
| On-line Update | 5 |
| Complex Processing | 1 |
| Reusability | 0 |
| Installation Ease | 3 |
| Operational Ease | 2 |
| Multiple Sites | 0 |
| Facilitate Change | 2 |
| Sum | 27 |

**Table 11‑3**: Influence Multipliers

As demonstrated by the table 11-3, the total value of the Influence Multiplier tuned out to be of twenty-seven.

After getting an adjustment factor, we calculated the influence multiplier with the following formula:

**Influence Multiplier = (27 \* .01) + .65 = 0.92**

Multiplying the Influence Multiplier by our Unadjusted Function Point total, we obtained our Adjusted Function Point total.

**Adjusted Function Point Total = 0.92\* 182 = 167.44**

Lastly, we use the Jones First Order estimation procedure to calculate the time it would take to develop a project using the established influence multiplies and unadjusted function points. The Outreach Storage System falls under the “Systems” category, which is analyzed in Best Case, Average Case and Worst Case scenarios, the calculations are displayed in the following table.

**Duration = 172.90.45 = 10.01 Calendar Months**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Best Case** | **Average Case** | **Worst Case** |
| **Adjusted Function Point** | 167.44.39 | 167.44.42 | 167.44.45 |
| **Total** | 7.36 | 8.59 | 10.01 |

**Table 11‑4**: Jones First Order Estimation

The data in Table 11-4 shows that at our very best the project will take approximately seven and a half months to complete and in the worst case scenario it could take up to 10 months.

The second estimation model is the Simplified Hybrid Approach. We decided to take this approach because using function points for estimation is typically more accurate. The following table calculates the estimated the lines of code of according to our function points based on the QSM Function Points Languages Table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Type** | **Language** | **Approximate LOC/ Function Point** | **Total** |
| Number of inputs | HTML | 34 x 72 | 2,448 LOC |
| Number of outputs | HTML | 34 x 21 | 714 LOC |
| Inquiries | SQL | 21 x 4 | 84 LOC |
| Logical internal files | Java | 65 x 53 | 3, 445 LOC |
| External interface files | Java | 65 x 10 | 650 LOC |
| **Total** | 7,141 LOC | | |

**Table 111‑5**: Simplified Hybrid Approach

The data in Table 11-4 describes how the SLOC calculations for the different function points were calculated, based on QSM Function Points Language Table. Producing a result which indicates that at our very best the project will take approximately seven and a half months to complete and worst case scenario it could take up to 10 months.

After calculating the estimated LOC we now need to calculate the effort, which is calculated based on the COCOMO simplified formula, which multiplies the KSLOC by a factor of one point four:

**Effort = 1.4\* 7.141 = 9.99**

Finally, we made use of the Rule of Thumb formula to calculate the duration of the project:

**Time = 3.0\* 9.991/3 = 6.46 months**

The third estimation model that we decided to use was the Sanity Test method by Weiss and Wysocki. For this sanity test, we used the results gathered from the Simplified Hybrid Approach, hoping to get similar results so that we can get a more accurate estimation. To calculate sanity test, we need to make use of an optimistic duration which in this case is the result that we obtained by the Hybrid Approach (6.46 months). For the nominal duration, we used the best case obtained in Jones’ First Order Model (7.36 months). The third estimate we used is the pessimistic duration, also obtained by the Jones’ First Order method (10.01 months).

The following formula represents the Sanity Test calculations and results.

***E= O + 4M + P / 6***

***E= (6.46 + 4(7.36) + 10.01) / 6 = 7.6 Months***

The sanity test results indicate that according to our calculations, we would need seven and a half months to finish the Outreach Storage System. Our team will be able to implement the most critical and high importance requirements within this period. Low priority items will be developed towards the end of the cycle, to ensure that critical items are fully developed in the given time frame.

# Future Items

All of the requirements listed below will be implemented as time and schedule permit.

## 12.1 Crate Containment Tracking

### 12.1.1 Description: The system shall track items contained within each individual crate.

### 12.1.2 Constraints: Cost and Time. The budget that we have currently allocated is not enough to account for costs related to individual crate tracking, such sensors or other technology types. Also the time will not be enough for the team to implement this functionality, therefore the team agreed on implementing a fast mechanism to track the items, trying to increase the performance of the system.

## 12.2 Mobile Application

### 12.1.1 Description: The user shall be able to manage the Outreach Storage System using a mobile application.

### 12.1.2 Constraints: Time. The time that we have for the completion of the project, is only enough to develop a web application. If the team happened to develop a mobile application, the project plan will be in jeopardy causing some tasks to be incomplete. The team has decided to develop a responsive web application, so that it can be launched in a mobile browser.

## 12.3 Locating Crate Inside the Storage Room

### 12.1.1 Description: The system shall be able to provide a relative location of a crate inside the storage room.

### 12.1.2 Constraints: Cost. With the current budget, it is impossible to buy three extra antennas and a more expensive reader to support them.