Household Inventory Management System

Isaac Tucker

CST-452 Capstone Project Proposal

Grand Canyon University

Instructor: Michael Landreth

Date: 05/15/2022

**ABSTRACT**

This project will be a cloud deployed application for household inventory management. Users should be able to register, log in, create/read/update/destroy household listing and add/read/edit/delete items within each respective household. Households can be shared among users for collaborative inventory management. Items can be flagged for donation.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 04/03/2022 | Isaac Tucker | Initial draft for review/discussion |
| 05/13/2022 | Isaac Tucker | Revision for final submission. |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments**  No Feedback. |

|  |
| --- |
| **Overall Instructor Feedback/Comments**  No feedback. |

**Integrated Instructor Feedback into Project Documentation**

Yes  No, no feedback was provided.

**Project Approval**

Professor Amr Elchouemi

Table of Contents

[Project Overview and Project Objectives 4](#_Toc99801109)

[State the Problem and Background 4](#_Toc99801110)

[Christian Worldview 4](#_Toc99801111)

[Project Objectives 5](#_Toc99801112)

[Challenges 5](#_Toc99801113)

[Benefits and Opportunities 5](#_Toc99801114)

[Project Scope 6](#_Toc99801115)

[Project Success Measures 7](#_Toc99801116)

[Project High-Level Solution 8](#_Toc99801117)

[Introduction 8](#_Toc99801118)

[Solution 10](#_Toc99801119)

[Project Controls 12](#_Toc99801120)

[Project Cost and Schedule 13](#_Toc99801121)

[Appendix A – References 14](#_Toc99801122)

[Appendix B – Copyright Compliance 15](#_Toc99801123)

# Project Overview and Project Objectives

## State the Problem and Background

This writer is a homeowner and often has difficulty keeping track of the items within his household; there have been times when multiples of the same item are obtained/purchased and money is wasted as the fact that those items are already in one’s possession is overlooked. For that purpose, I propose a household inventory management system where a user can list households and keep a running inventory of items for each household. The user might then share their household inventory listing with others who live at the household for collaborative contribution to the inventory list. Moreover, in an effort to help reduce clutter, the inventory listing will have features to indicate items needing to be purged and integrate/reference local community organizations in need of donations.

## Christian Worldview

The product this project would provide is an inventory management system; anyone and everyone will obtain items in their life that could and should be tracked. This will provide users with a convenient means to see what items are in their possession and a means to better organize those items (and their life, by extension). In this way if another party is, perhaps, in need then a user of this product can see if they have the capability to assist that party and provide needed items (i.e. a neighbor is in need of a shovel and a user might see they own 4 so they either donate/lend excess items). In this way a user of this product can embody the Christian ideal of loving their neighbor, as Jesus would, and clearly see and act upon their ability to share what they have for the betterment of all.

Moreover, the product (inventory management) would contain the ability to flag certain items for donation and present a public facing report of items for donation. In this way organizations designed to help those in need (i.e. goodwill, red cross, Arizona homeless shelters, Terros community support, etc) can make use of these listings and reach out as needed.

## Project Objectives

* Register/Login User
* Allow User to create/read/update/destroy household(s)
* Allow User to create/read/update/destroy item(s) within a household(s)
* Allow User to share household inventory items with others
* Allow User to flag certain items for donation

## Challenges

* The most prominent limitation for this project is time; the courses within GCU’s nontraditional modality are accelerated, and time is limited to the scope of the classes themselves.
* The intent is to have the product (household inventory system) available online but funding for online deployment is limited. So, this developer is unsure of his financial capability to actively host the product for utilization.
* Due to restrictions of GCU course availability this developer has yet to take and learn about cybersecurity fundamentals. As such this developer is unaware of some security practices that may be best suited for an online application.
* The Software Development program of study did not have a large focus on GUIs. Thus the GUI for this application will likely be simple and unattractive.

## Benefits and Opportunities

The internet is currently saturated with online marketplaces and social networking sites but there is a lack of websites for people to log the items in their possession and have the capability to share/donate items in excess, social SHARING site versus social networking site. When people have unneeded items they typically end up leaving said items on the lawn for whomever to take or they might, if they have time, host a yard sale. The reality is that those in need are typically busy to make ends meet and often go without as they do not have a means to express need on a platform that can also help resolve that need. The benefit of this product is that it will provide an opportunity for users to better organize their household and provide an opportunity to share inventory logs and items flagged for donation. Moreover, it can help direct those same users to organizations that can help connect excess items with those in need.

# Project Scope

This project will be a cloud deployed application for household inventory management. Users should be able to register, log in, create/read/update/destroy household listings and add/read/edit/delete items within each respective household. Households can be shared among users for collaborative inventory management. Items can be flagged for donation.

|  |  |  |
| --- | --- | --- |
| Stakeholder Name | Role(s) | Responsibilities |
| Isaac Tucker (self) | Developer | Plan and develop the application and all required parts |
| Homeowners | User | Add/Edit/Del households/items, mark items for donation. |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Establish NodeJS Backend for User Registration/Verification, Household Create/Read/Edit/Delete, Item Create/Read/Edit/Delete, Mark item for donation, Share household. | NodeJS | Pending | 10 | No Cost | 04.01.22 | 04.24.22 | 05/15 | 05/05 | Past Class notes |
| 2 | Establish MySQL DBMS to store Users/Households/Inventory | MySQL | Pending | 4 | No Cost | 04.01.22 | 04.24.22 | 04/23 | 04/23 | Past Class notes |
| 3. | Create React front-end for users to interact with application | React  Bootstrap  Axios | Pending | 10 | No cost | 04.01.22 | 04.24.22 | 05/15 | 05/10 | Past Class notes |
| 4. | Deploy NodeJS cloud service (Probably Heroku) | Heroku | Pending | 10 | Unknown Cost | 04.24.22 | 04.31.22 | 05/15 | 05/11 | Heroku FAQ/Help |
| 5. | Deploy React front-end to cloud service (Probably Heroku) | Heroku | Pending | 10 | Unknown Cost | 04.24.22 | 04.31.22 | 05/15 | 05/12 | Heroku FAQ/Help |
| 6. | Establish JawsDB MySQL hosting for DB | Heroku JawsDB | Pending | 4 | Unknown Cost | 04.24.22 | 04.31.22 | 04/23 | 04/23 | Heroku FAQ/Help |

# Project Success Measures

|  |
| --- |
| Project Completion Criteria |
| 1 – Users can register/login into application |
| 2 – Users can create/read/update/delete household listings |
| 3 – Users can create/read/edit/delete items from household |
| 4 – Users can flag items for donation |
| 5 – Users can view list of items for donation |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 | Users have time to individually log items within household | The task of data entry is a tedious one and not all users may have the patience for this. | Assumption |  | 04/02/2022 |
| 2 | Users are truthful | Item listings will be truthful and not embellished or outright lies | Assumption |  | 04/02/2022 |
| 3 | Household owners have internet | Users will need internet to access application | Assumption |  | 04/02/2022 |
| 4 | Development Timeframes are limited | This length of these accelerated classes are limited and thus time to implement all intended user stories is also limited. | Constraint |  | 04/02/2022 |
| 5 | Cloud Deployment Costs | Costs to deploy applications to the cloud have real world scaling costs and said costs may be outside the scope of the development budget. | Constraint |  | 04/02/2022 |

# Project High-Level Solution

## Introduction

To achieve the intended project, Household Inventory Management System, we will make use of a database management system for data persistence. Information to and from this database will be communicated with a back-end REST service. Then a front-end will provide an interface for users to make use of the REST service and make use of the database. These three components will make up the application itself. The needed application might look something like the following diagram.

Diagram

Description automatically generated

The intention of the REST service will be to decouple the logic for the database interactions from the view itself. This will allow for easier implementation of other GUI front-ends if the application grows onto other platforms as it will provide a centralized means of communicating with stored data.

Then, to state the application objectives in user story format:

**Epic:**

As a user I want a system where I can track my various items that I have within my household(s).

**User Stories:**

* Register: As a user I want to be able to register with the application so I can use the application and have a unique account to associate my household entries with.
* Login: As a user I want to be able to log into my account to access my households.
* Add household: As a user I want to be able to make different household listings so I can separate items based off different locations.
* Delete household: As a user I want to be able to delete households in case I no longer need the inventory list.
* Update household inventory: As a user I want to be able to update my household information in case I made a mistake or details change.
* View households owned/managed by user: As a user I want to be able to view the households that I am managing.
* View household items for households owned/managed by user: As a user I want to be able to view the inventory items that are currently in a select household.
* Add Items to household: As a user I want to be able to add items to a household listing.
* Delete Items from household: As a user I want to be able to delete items from a household listing.
* Update Items in household: As a user I want to be able to update item information within a household.
* Flag for donation: As a user I want to be able to flag items for donation.
* Invite Collaborators: As a user I want to be able to invite other users to also add to my households.

## Solution

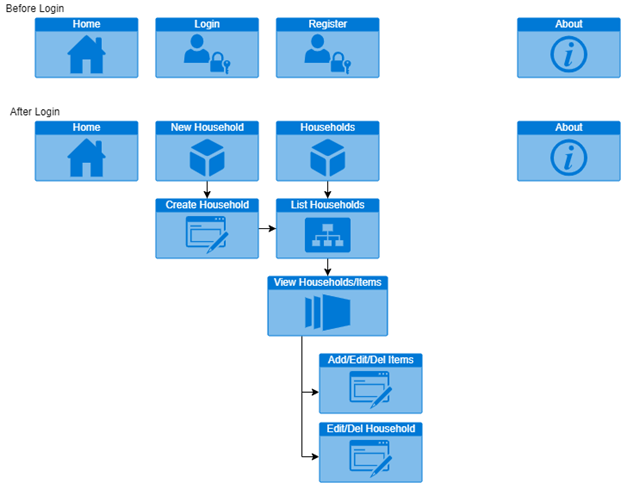
DBMS: I expect that the DB will be deployed on Heroku using the JawsDB MySQL hosting. An initial ERD of the DB may look like…

Diagram

Description automatically generated

The main intention here with this database structure is that one household will have many items. Then there is an intermediary table to hold user and household associations. This way households can be shared with other users and stored within this table to allow collaboration with household item data entry. The expectation is that, upon household creation, a household\_users entry will also be created to affiliate the new household ID with the creating user ID. Then if additional users are invited to collaborate, additional user/household pairings will be made.

An initial sitemap for the intended front-end might look like…



So, before users are logged in they will be restricted to the homepage, a login page, a register page and an application about alert. Once the user is authenticated then they will gain access to the core part of the application which allows them to create households, view their households they have access to, view the households themselves along with the items inside that household, add, edit and delete items within a household, edit and delete households themselves.

# Project Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Management | | | | |
|  | **Risk Probability** | **Risk Impact** |  |  |
| **Event Risk** | **(high, medium, low)** | **Risk Mitigation** | **Contingency Plan** |
| Cloud unavailable | Low | Users unable to access application | Multiple deployments on multiple cloud services. | Make application accessible on multiple cloud deployments |
| Programming Errors crashing servers | Medium | The application might crash making application inaccessible. | Run QA testing on application to ensure bugs at kept minimal and ensure logs are present to keep record of bugs when they occur. | Use a site up tracking service to alert development staff if servers ever go down for some reason. This will allow development team to restart servers asap. |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues Log | | | | | | | | |
| **ID** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | Navigation between certain components | Minimal | Research better routing practices | Isaac Tucker | *Minimal* | *05/09/22* | *05/20/2022* |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change Control Log | | | | | | | | | |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 | None at this time. |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Roles and Responsibilities | | | |
| Name | Team | Project Role | Responsibility |
| Isaac Tucker (Self) | Developer | Developer | Everything. |
|  |  |  |  |

# Project Cost and Schedule

1. Expected costs:

|  |  |  |  |
| --- | --- | --- | --- |
| Expected Cost Breakdown | | | |
| ID | Service | Purpose | Expected Cost |
| 1 | Heroku JawsDB Dyno for MySQL DB | Needed for MySQL DB hosting for overall application data persistence. Will start at free tier: 5 MB storage, 10 connections. Next tier is $10/month: 1 GB storage, 15 connections. <https://elements.heroku.com/addons/jawsdb> | Variable: Free to $47,000 month. |
| 2 | Heroku NodeJS REST Hosting | Needed for hosting of back-end REST service to communicate with above DB for data persistence. Will start at hobby tier: 550-1000 dyno hours/month. Likely may need to bump to first production tier of $25/month. <https://www.heroku.com/pricing> | Variable: Free to $250+/month |
| 3. | Heroku React Front End Hosting | Needed for hosting of front-end React application for overall application access Main entry point for users. Will start at hobby tier: 550-1000 dyno hours/month. Likely may need to bump to first production tier of $25/month.  <https://www.heroku.com/pricing> | Variable: Free to $250+/month |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project Schedule and Timeline | | | | | | | |
| ID | Task | Status | Effort Hours | Start Date | Planned Completion | Estimate to Completion | Actual Completion |
| 1 | Set-up DBMS Schema and deploy to Heroku | Complete | 6 | 04.01.22 | 04.24.22 | 6 hrs | - |
| 2 | Build NodeJS back-end service to communicate with DBMS. | Complete | 10 | 04.04.22 | 04.24.22 | 10 hrs | - |
| 3. | Deploy and Test NodeJS service (Heroku/Postman). | Complete | 6 | 04.10.22 | 04.24.22 | 6 hrs | - |
| 4. | Build React Front-End | Complete | 10 | 04.17.22 | 04.24.22 | 10 hrs | - |
| 5. | Deploy and Test React Front-end | Complete | 6 | 04.20.22 | 04.24.22 | 6 hrs | - |
| 6. | Isolate and correct any bugs within application | Complete | 6 | 04.21.22 | 04.24.22 | 6 hrs | - |
| 7. | Refactor/Improve any components of application needing improvement. | Complete | 6 | 04.24.22 | 04.24.22 | 6 hrs | - |

# Appendix A – References

*None at this time.*

# Appendix B – Copyright Compliance

Bootstrap: <https://github.com/twbs/bootstrap/blob/v5.0.2/LICENSE>

Bootstrap is a library to help with responsive display elements for the GUI of an online application. It helps with sizing and format to ensure a response display across devices of various screen sized. Thus this bootstrap library will be used within the HTML code to help coordinate the view display to the user.

The MIT License (MIT)

Copyright (c) 2011-2021 Twitter, Inc.

Copyright (c) 2011-2021 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

React-Axios: <https://github.com/sheaivey/react-axios/blob/master/LICENSE>

React-Axios is a library used for the react front-end to communicate with the NodeJS REST back-end. It allows for HTTP method requests to be passed between the components of the application.

Copyright (c) 2016 Shea Ivey

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

NodeJS / ExpressJS: <https://github.com/expressjs/express/blob/master/LICENSE>

NodeJS / ExpressJS is the framework that the back-end REST service will utilize to communicate and interact with the MySQL database.

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>

Copyright (c) 2013-2014 Roman Shtylman <shtylman+expressjs@gmail.com>

Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the

'Software'), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY

CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.