Where's My Stuff
Milestone 2

CS 462
Winter Term 2021

Group 24
Michael Friesen, Shawn Kilbride, Thien Nam

Abstract

Where's My Stuff looks to solve the issue of losing items in cluttered workshops and extensive inventory systems through a database system of all inventory items accessed via either text or voice commands. The solution has been implemented using a MySQL database accessed by backend APIs using an Express Node.js server. These APIs are used in both the voice processing application through Google Assistant and the frontend textual website created using React. These pieces all mesh together to create a seamless user experience in accessing, viewing, and modifying their inventory systems.

Table of Contents

1) Background:

1.1) Current Conditions

The constant struggle with searching for specific items in storage systems has led to the need for a simple system to locate the item through voice commands that anyone is able to use. Where's My Stuff looks to solve this problem through integrating a database and voice processing software with a voice recognition system to quickly identify where items are located. Currently, the team is working on researching different softwares and interactions to devise a solution that seamlessly integrates all parts. We have decided to use a MySQL database with a NodeJS backend that will help handle incoming requests as well as process audio commands.

1.2) Existing Systems

Prior to this project there has been some work done on the side of more precise inventory tracking, leading to specialized bins being created to track the amount of inventory. Additionally, there have been inventory systems that have been developed for large scale manufacturing and businesses to keep track of merchandise inventory but those are primarily focused on enterprise. As for personal inventory with a focus on voice interaction, there doesn't seem to be anything that is publicly available for use. This existing system will be useful in understanding how to better track the inventory through these bins and observe the interaction between the code and hardware of the bins.

1.3) Problems

We have made a large amount of progress in the development of this product so far. We are still running into issues with the development of the voice processing software that will enable voice commands to be used. We need to research how to connect the voice software that is currently in development to the backend API application to gain access to the API queries on the database.

1.4) Pertinent History and Stakeholders

In researching for this project, several similar products have been discovered such as Smart Bin Solutions. There are several bin systems that track inventory through a similar means with physical bin hardware. However none of these options seem to have effective search capabilities such as with voice commands as this project plans to implement. The use of voice recognition software has led to research into different means such as a simple phone microphone or using Google Home or Amazon Alexa technologies. The last two technologies have the capability of adding functionality to their voice

processing and thus may provide a solid solution to the voice recognition issue. These are relatively new technologies that will 0need to be researched more to understand the ease of implementation.

This project has come about from Mark Reed, a retired HP employee looking for innovative solutions to inventory management systems. As the project continues, the assistance and advice from professors or professionals may be consulted for advice on issues, but the current stakeholder and project lead remains Mark Reed.

1.5) Terms, Concepts and Definitions

Voice Recognition Software: This is the software and hardware we will use to transmit audio and process it such as Alexa or Google Home combined with our created software.
MySQL: a SQL language for creating, manipulating and handling a relational database.
NodeJS: A JavaScript framework used to create network applications for handling requests.
React: A Javascript library used for building user interfaces
API: Stands for application programming interface, and is an interface that defines interactions between multiple software intermediaries
Database: Organized collection of data that is stored and retrieved with from the API

---

2) Vision:

2.1) Vision

This project will work to solve the issue of spending an unnecessary amount of time searching for specific items. Where's My Stuff provides a solution that allows users to quickly and efficiently locate inventory within their storage systems that allows them to streamline their production process and save time/money. Life for individuals and industries alike will be greatly improved by the ability to much more quickly acquire what they need and track the amount of inventory present. In creating this product there is a set of prioritized goals that will be met to deliver the most complete project. First and foremost is a database that stores the inventory and a voice processing system that allows users to access the database. The user will also be able to modify inventory and find availability in their system through various voice commands. One of the final goals that may or may not be finished by the deadline will be notifications for low inventory and other various statistics and easy exportation of inventory information. There may be potential for other features given time constraints and feasibility.

2.2) Central Hypotheses

Growth hypothesis:
Users will be attracted to this product as the technologies used are relatively new and growing quickly. Voice technology is a growing industry that reinforces the idea of a “smart home” to many users and helps to further improve the efficiency and ease of daily life. The simple and intuitive design will attract users to this new form of inventory management.

Value hypothesis:
The benefits of Where’s My Stuff lies in the saved time from being able to quickly locate items in large inventory systems. The system will be intuitive and easy to use so as to allow users to come on board and quickly get value from the product. A limited amount of technology and setup is necessary thus making the product more user friendly and valuable.

2.3) High-level Requirements

Functional requirements:

- The system will store a database of information about the location of items in containers.
- The user can interact with the system via the web application.
- The system will interact with the user via voice commands and audio feedback.

Non-functional requirements

- The system should remain decently organized.
- The system should return accurate information based on the contents of the database
- Quickly processes voice commands and accesses information

---

3) Prioritized Project Constraints:

3.1) Time

At this point in the project, we have made a considerable amount of progress in completing the alpha version of the product. We have mostly completed the backend, database, and frontend web application for our project. The next big component to add is the voice recognition component. Given the current status of this piece and the time remaining, we hope to get it integrated with the backend APIs before the end of winter term. After getting the voice component integrated with Google Assistant, will begin testing and fine tuning the commands, as well as adding more features into the web application.

3.2) Resources

From researching potential solutions to this problem, we have come across numerous resources that will be useful. One constraint that hasn't been thoroughly researched yet is the need for hosting our product and a database in the cloud. This may require small financial resources if using services such as Amazon Web Services or Google to host our program. Other resources fall under the category of personal development items like laptops or voice assistant technology, however most members have access to these resources.

3.3) Scope

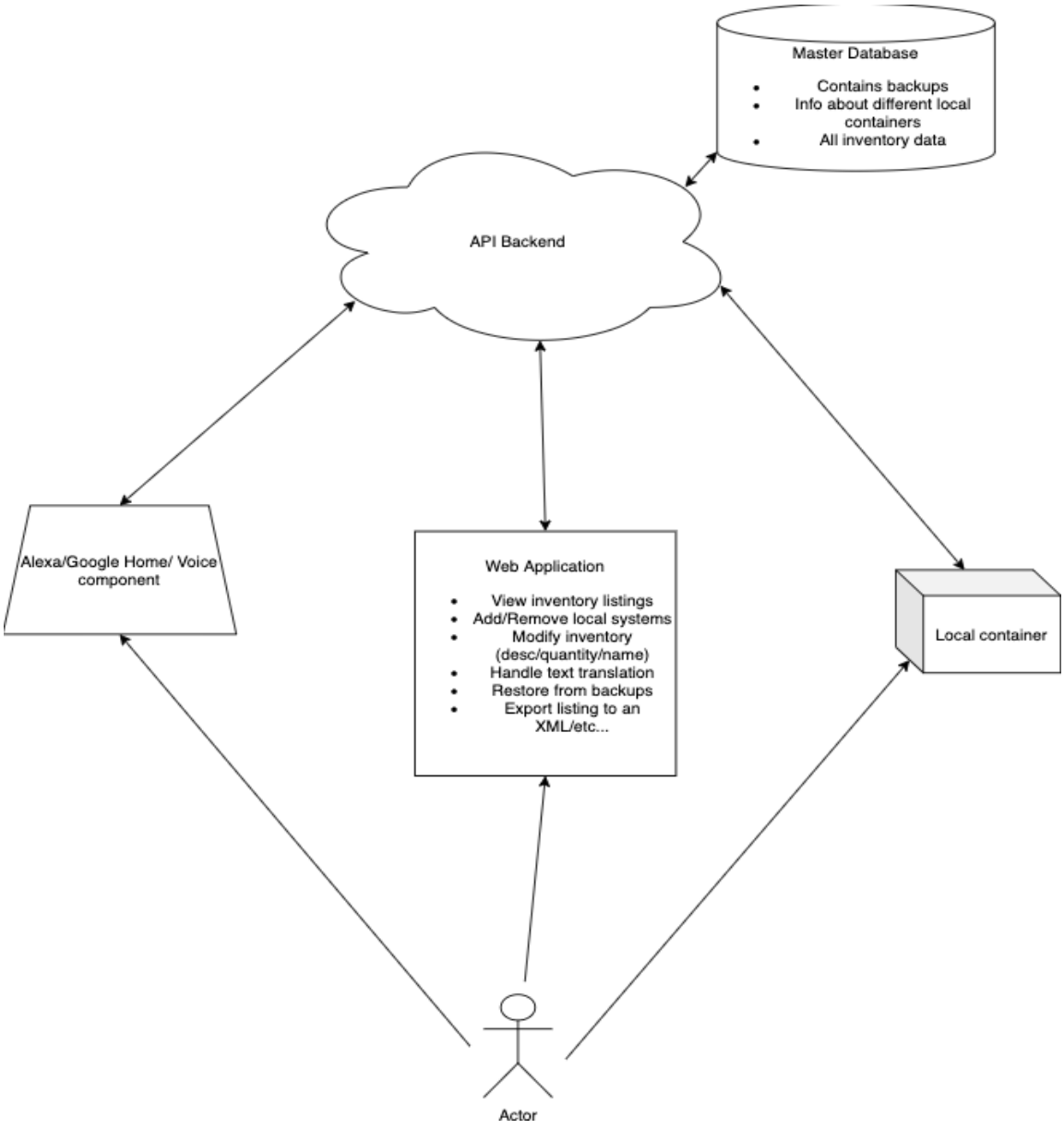
In order to create the minimal viable product we will need to create a web application that shows all the existing items and has an ability to add items, as well as integrate with a voice assistant (Google Assistant) which should be able to handle basic commands to our API backend containing the information. Voice commands will be used to access these APIs and get data, but the voice commands need to be interpreted from our program. These tasks should be feasible to complete in the time of the project given our current level of research and development. Given our resource constraints, it will take some research and time to understand how to host our program on the cloud and understand the resources required for this constraint.

---

4) Scope:

4.1) Process Flows:

- (Voice) User wants item -> asks application -> application returns information about item
- (App) User wants item -> searches for item -> application has information about item
- (Voice) User takes item -> decrement item in app -> application has an accurate count of items
- (Voice) User has new item -> tells application about new item -> application adds new item to database
- (Voice) User loses item -> tells application -> application deletes item
- (Error Handling) User says something -> application doesn't understand -> application asks for clarification.
- (Error Handling) User asks for item -> Item not in database -> application says they don't have a record of that item.



4.2) User Stories:

- As a user I need to access items in the database with my voice so that I can remain hands free and improve my workflow.
- As a user I need to locate items in my workshop with simple voice commands so that I can quickly find what I need.
- As a user I need to be able to use voice commands to show what items are in my system so that I can see what all I have available.

4.3) Supported Voice Commands:

- **Minimum product**
    - Add Item: “I have a ____.”
    - Remove Item: “I don’t have any more ____.”
    - Increment Item: “Add x quantity to ____.”
    - Decrement Item: “Remove x quantity from _____.”
    - Find Item: “Where’s my ____?”
    - Create a new category: “Add a new category called ____.”

- **Planned commands**
    - Filter Item: “Show me all ____.”
        - By category: “Show me all ____ categorized by ___.”
        - By container: “Show me all ____ in ____.”
    - Find space for an item: “Where is the nearest empty container?”
- **Stretch Goal Commands**
    - Low items: “What do I need to restock?”
    - List all containers: “What are all of my containers?”
    - Export to csv: “Export my stuff.”
    - List all items: “What all do I have?”
    - Reserve container: “reserve container ___ for ___”

---

5) Iteration Plan and Estimation:

Task	Time Estimation
Beginnings of research and layout	1 month
Create skeleton of database, web application	1 month
Add basic voice processing for accessing database	6 weeks
Add sample data and add more commands	6 weeks
Testing and fine tuning	1 month

Minimal viable product finished	At 6 month mark
Add inventory notifications	2 weeks
More advanced filtering, locating, inventory management commands	6 weeks
Final testing and finishing touches	1 month

---

6) Alpha Functionality

6.1) Backend + Database (Michael)

My name is Michael Friesen. I am an undergraduate studying computer science. I am mostly interested in backend and databases work. I was tasked with creating the database and backend application to supply APIs that query the database. I was able to get a MySQL database up and running with our desired schema and connect backend APIs to the database. The backend application is written in a Restful format with APIs to do CRUD operations with the different tables and categories of the database. The schema has changed slightly with tables added and modified, and changes to the backend and APIs have been reflected with modified/more APIs and changes to the schema file. I have also been able to host the frontend, backend and database on Amazon Web Services using an EC2 and RDS instance to host them. This allows the text website to be exposed publicly for testing. Continuing on, I will be helping with the voice processing and fixing bugs as they appear.

6.2) Frontend + Helping out backend / database (Thien)

My name is Thien Nam. Finishing up my final year at Oregon State, I am also working part-time at Expensify, working across the tech stack to fix bugs and build out functionality for expense management. My main task this milestone was building out the web application and having it handle various features for viewing, modifying and adding items, containers, and the relationship between the two. I created a React front-end and had it list all the items along with where they are stored, and added the ability to modify and delete the information corresponding to the items and containers. Most of the functionality that we set out to have in the front-end is complete, with the exception of the adding items to containers, item categories, voice commands, and just making it look good. I have implemented some dummy commands for voice processing that can extract a target item from a voice command, and just need to run a wrapper to send the commands to the proper endpoints. Additionally, I've helped oversee the development of the backend and database with Michael, and added a few routes when needed to connect information to the front-end from the API.

6.3) Voice Processing + UX (Shawn)

My name is Shawn Kilbride. I am an undergraduate at Oregon State University majoring in Computer Science. I also work as the Head of IT at Changing The World Inc, a 501(c)(3) non-profit organization. My task was to allow for the user to utilize voice commands to interact with the database. This is the primary and titular method for user interaction with the database. I created and developed a

google actions project to implement this functionality. The project is still a work in progress, but currently has dialogue laid out for the user to request to find items or containers, add items or containers, or remove items and containers from the database. It can reliably hear and understand what the user is saying in most situations, and has the ability to ask for clarification if it does not understand. The implementation of this project is stored on google's actions console where the project is developed. The minimum product for this branch of the whole project is almost complete, and will soon be ready for integration with the rest of the application.