University of Arkansas – CSCE Department
Capstone II – Final Report – Spring 2021

InventorME

Floyd Brown, Esteban Duran, Anthony Lopez-Guerra, Jaden Martin, Luke Miller, Benjamin Thedford, Chase Zimmerman

Abstract

Disorganization is a hard problem to solve that almost everyone faces. It can lead to people wasting time looking for lost items. If those lost items were properly organized, then the time spent looking for them could have been spent on something more productive or enjoyable. The objective of InventorME is to solve this problem with an application that helps to manage personal items. Furthermore, the objective is not only to solve this problem, but to prevent the disorganization from occurring in the first place.

The approach of InventorME is to have an inventory application that is accessible and flexible to the user. Its focus on flexibility is based around its ability to allow the user customization. In the case of accessibility, it is cross-platformed: the user can access it from any device, anywhere, at any time. To put it simply, the application emphasizes ease of use. The significance of InventorME is that by solving this problem, the user can be more organized. Depending on the user, InventorME can help them get their life back, and can aid them in their personal pursuits, and personal happiness.

1.0 Problem

The problem of disorganization is one based around the mismanagement of items. This involves the misplacement of an item. This misplacement can cause confusion and can lead to an individual searching for an item for an extended period of time. In some cases, the individual may not ever find the item. If the individual does not have a duplicate of the item (which often is the case), this makes losing said item even more costly.

This problem is important because certain items, such as car deeds or marriage licenses, hold a lot of weight in a person's life. If an individual misplaces or loses an item that is extremely valuable, it can cost that individual an extensive amount of money or stress to deal with the lost item. In some instances, it can even put their financial situation in jeopardy. If the item was something more personal, such as family heirloom, the misplacement of the item could potentially damage an individual's relationship with another person. This can bring about a variety of consequences, depending on the relationship; many of which can cost a heavy toll on an individual.

The impact of not having a solution to this problem is quite simple: the problem will continue happening. In other words, the status quo (or, what is normal right now) will continue going on. As illustrated, for some individuals, this can potentially be devastating to them. To put it blatantly, if the problem is never solved, then the consequences that arise from the problem will continue to exist.

2.0 Objective

The objective of this project was to create an application that solves the problem of disorganization. InventorME accomplished this by allowing the user to easily manage the items that they feel need management. In other words, the user inputs the items that they want the application to help them manage. The objective is accomplished by having the user input their item into the application, add any necessary details (such as an image, name, worth, etc.) and then categorize the item the way that they see fit. The item, and the details that compose that item, are accessible across multiple devices, that way the user can edit or add new items in whatever medium is most convenient. Finally, the data that the user inputs for their item is stored securely to prevent the loss of data.

3.0 Background

The main justification for the project was that it is hard for people to keep track of all the information related to their belongings. With this application, the user can easily retrieve all the information related to their items. This project not only focused on creating a solution to keep an inventory of the user items but also heirlooms, personal notes, and overall statistics for their personal projects. What we wanted to achieve was the flexibility for the application to adapt to the needs of the user’s needs, and we believe we accomplished this. There are some other personal organizing applications already. The problem that our project fixes, that the other applications don’t, is that they do not give the user the flexibility of adding and viewing much more specific data. In some situations, the user could be looking for info or photos on a project they did previously. This situation could be incredibly frustrating, with the use of our application the user could have a reference to an important file location making it easier for them to retrieve info on their project. This not only creates a more pleasant experience when searching for a particular object or information, but it is also going to help the user not spend hours searching.

Having something that reminds you of the location and key information that you need is a powerful tool to organize, not only personal items but also, professional projects and even payments. Our main goal is to create a database that is easily accessed. Therefore, we decided to not only work on a mobile application but also create a website in which you could see the same information as on your mobile phone. This is going to make the application have a broader audience since it might not be just for personal use specifically, but it could also be used by small companies.

3.1 Key Concepts

InventorME is available in three different formats: An iOS application, an Android application, and a web application. Each application provides the same service on all the different platforms, so that a wide variety of users can take advantage of the power of these tools. There is a database backend that stores and backs up the information for users that allows the retrieval and modification of data. As well as an S3 storage bucket for the Image storage/backup. The iOS and Android applications were built using Expo and React Native, which maps code to the platform that the application is running on. This allowed us to only write code once, which was then mapped to both iOS and Android using React Native. We initially planned to use XCode for the iOS app, and Android Studio for the Android app, but React Native provided more flexibility and interoperability.

The web application was built using ReactJS, which is the basis for react native. This ensured that the styling of both the mobile and web applications were similar. We were also able to share some code between the different platforms due to the similarity in the two languages. While we planned to use simple HTML and JavaScript, we decided to use ReactJS for consistency between platforms.

The backend was built using Amazon web services, which is the industry standard for most backend servers used today. We used a serverless design, meaning we used different modules/services for the different tasks we had, instead of a jack-of-all-trades server running 24/7. The services we used were Cognito for the account management, API Gateway for the REST APIs we built, Lambda for the functions that ran those APIs, RDS for the database storage, and finally we used MySQL for the database. The services seemed well kept and well documented, meaning the learning curve was lower. They also allowed for compatibility for the platforms we had in mind, which was very important for our cross-platform goal.

3.2 Related Work

One of the most well-known mobile applications for personal inventories is called Sortly [1]. Sortly is an application designed mainly to have an inventory separated in folders of your items. This application currently has an average five-star review with approximately 5800 reviews. Sortly is a good application to create an inventory of all your items. It features a webpage in which the user can access all the information from the application. It also features a pro version in which the user can pay to get extra futures. It is also able to store a photo of the item, that way the user can know exactly what is being looked for, one of the newest features of Sortly is that the user can scan the barcode and it can autofill some of the information. It features a comfortable and easy to use user interface.

InventorME is going to be an information-based inventory available on the web, android and iOS. One of the main differences between InventorME and Sortly is that InventorME is going to let the user decide whatever they want to store, whether that is receipts, photos, manuals, or even monthly payment reminders. This allows the user to not only keep track of their items but also of digital information like photos. Even though most documents are stored digitally, some of life's most important documents require physical copies and users can sometimes spend hours searching for a birth certificate or a photo ID that has not been seen in over a decade. With how flexible InventorME is, the user could take a picture of the important document they want to save and write the location in the location attribute. Then when they need to find it, they search birth in their InventorME app and are given a direct location to where it was last seen. It might seem simple, but it is very effective, and could save tons of time to those who want to have a place to keep info on their belongings.

4.0 Design

There are three main pillars of our design that weighed heavily in our efforts to bring this project to life: minimalism, accessibility, & flexibility. Each of these pillars is as important as the next with each helping to build a strong architecture and a great experience for the end user.

With design in minimalist philosophy many think that this route is an easy one. Minimalism means simple, right? Quite actually the opposite. A minimalist design requires copious amounts of vision, planning and forethought. What we mean is that our design has a foundational backbone that is rigid and well thought out, does not use redundant code, and the user interface is not riddled with bloat that gets in the way of the user's experience. These design practices give the user what they want in a timely manner, built using well designed code. This is because the architecture has been thoroughly thought out, designed, and vigorously debugged, given that each team member knows precisely what they were to focus on. Minimalist design practices are also echoed throughout all the design guidelines that we are using for each set of user interfaces. For mobile we are using a hybrid of Apple's Human Interface Design Guidelines and Google's Material Design Guidelines. For desktop the InventorME's web page design follows Adobe's Web Page Design Guidelines which aligns well with the other two design languages. All three also have minimalist practices already ingrained in their design patterns. These minimalist practices allowed for a greater implementation of the other two pillars of our design -accessibility and flexibility.

With nearly everyone always having a smart device in their pocket with them it is key that no matter who downloads our app or visits our site, they can use it. This is where accessibility comes into play. Accessibility is an industry wide practice of enabling your programs and applications to accommodate the myriad of users who have a wide variety of disabilities. Disabilities such as vision and hearing impairments or people who need touch assistant features can have difficulty using a smart device. By opening the door to these users, it allowed for greater diversity and a better community around our application; communities are key to growth. It is worth it to promote a platform of fairness and equality, even if it only helps a single person. Enabling these features, such as text and layout scalability for example, also allowed for better future proofing of our application as many companies in the past have enabled new features in their OS that use existing accessibility features, such as dark mode on iOS that used attributes from accessibility for color inversion. All of this combines to give the users of our app the best experience possible, a practice also evident in our third pillar of design -flexibility.

Right now, the world is very different from how it was a year ago when Covid-19 hit, and the world was thrown in a whirlwind. Instability was the word many workers and students used when they had to transition to their work from home environments. One advantage many industries discovered is that flexibility can often lead to success. Companies that were flexible before the pandemic arrived took less of a hit in productivity and time to transition. Our group decided that ingraining flexibility into our design was not only extremely important in how the application was designed but also the means by which we developed it. In designing with the flexibility of the user in mind, InventorME needed to be available cross-platform (i.e., Android, iOS, and a web app) so that it is where the user wants to be, and we also used cloud synchronization (i.e., Amazon Web Services for the backend) so that the information is available when the user needs it. For instance, this allows for the user to log all their possessions on their laptop and add new ones on their phone later with ease. We also created an environment that allowed for flexible development. To enable this, we used a hybrid of waterfall and agile development practices; we had overall goals, but all the details were not locked down. That allowed us to add new functionality along the way, but the overarching functionality remained. Having the flexibility to build off our minimalist and accessibility goals put us in a really great place by the time we rounded out this project and left us with what we believe to be a fantastic product that is designed in and out with great intention and keen detail.

4.1 Requirements and/or Use Cases and/or Design Goals

Requirements:

- Backend System
- Database
- Cloud Sync
- Clients
- iOS
- Android
- Web App

Use Cases:

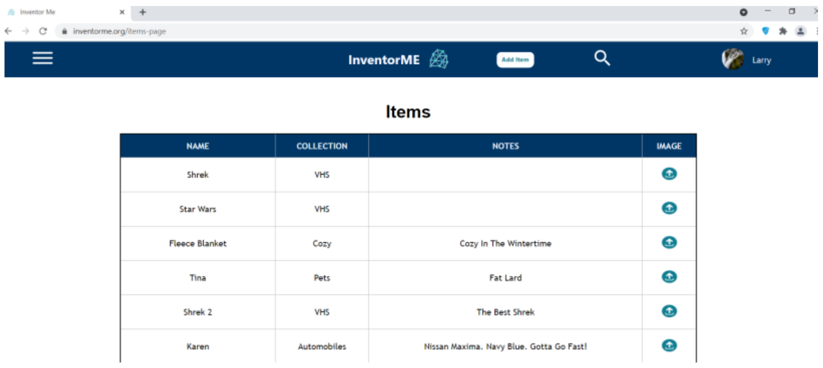
1. List Items
2. Documentation/Manuals For Item
3. Item Data
4. Item Organization
5. Track Location of an Item
6. Scan Item For Quick Input of Data
7. Data/Images Backup In Cloud
8. Track Value of an Item
9. View Collections
10. View Folders
11. View Statistics For an Item
12. Automatic Cloud Sync Between Devices
13. Allow Photo Upload of Items
14. Search Items

4.2 Detailed Architecture

The application, from its infancy, was created to be cross-platformed with iOS, Android, and web apps. In the case of the web application, it was straightforward. In terms of its architecture, the web application had a microservices architecture: that is, while it was being written, it was broken down into its smaller components, each were independent of one another. So, for example, the navigation banner at the top of the app is its own module; or in other words its own service that will work regardless of if the archived page is working properly. The reason that this route was chosen was because it allowed for easier scalability and it allowed the individuals who worked on it, both Floyd Brown and Jaden Martin, to produce quality software faster. The technologies used for the web application include React (a front-end API for JavaScript), HTML, CSS, and for the backend all the data was sent to an AWS API. The reason that these technologies were chosen were because it would be easier to implement and create the application using them. Additionally, it would be easier for the mobile and web developers to coordinate if one team needed help from the other, since the mobile app was developed using React-Native. An example of this was the Collections page. Esteban Duran designed and implemented it for the mobile application, and the web application team got stuck working on it. So, Duran was able to easily come over and help the web app team by starting the view. After implementing the basic functionality, Brown was able to finish out the Collections page.

The interface design for the web application was one of the most challenging parts for the web app team. However, it basically boiled down to this: except for the login/create an account pages, the web app was based around the navigation banner. This banner is what allows the user to successfully maneuver around the application. From there, the user can go on any page with data on it and edit that data with a simple click. In the case of collections, if the user wants to edit the data inside of a specific collection, they will have to click on the arrow displayed in the box of each collection. This takes them to a table where they can then edit the items inside of said collection.

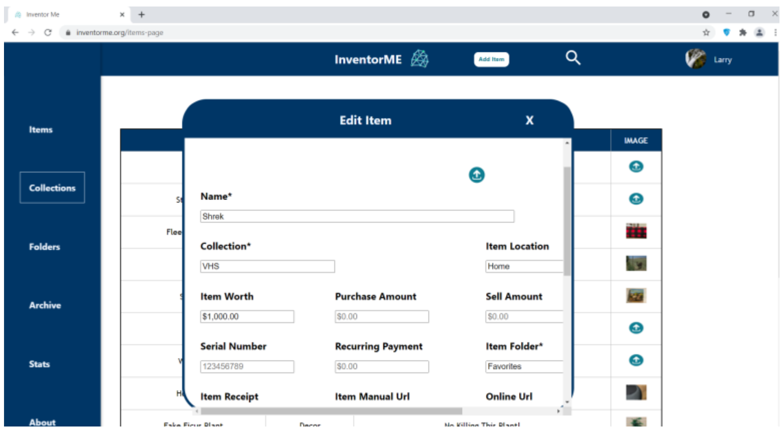
Navigation banner pictured:



Items

NAME	COLLECTION	NOTES	IMAGE
Shrek	VHS		●
Star Wars	VHS		●
Fleece Blanket	Cozy	Cozy In The Wintertime	●
Tina	Pets	Fat Lard	●
Shrek 2	VHS	The Best Shrek	●
Karen	Automobiles	Nissan Maxima. Navy Blue. Gotta Go Fast!	●

An example of editing an item inside of the web app:



Edit Item

Name*: Shrek

Collection*: VHS

Item Location: Home

Item Worth: $1,000.00

Purchase Amount: $0.00

Sell Amount: $0.00

Serial Number: 123456789

Recurring Payment: $0.00

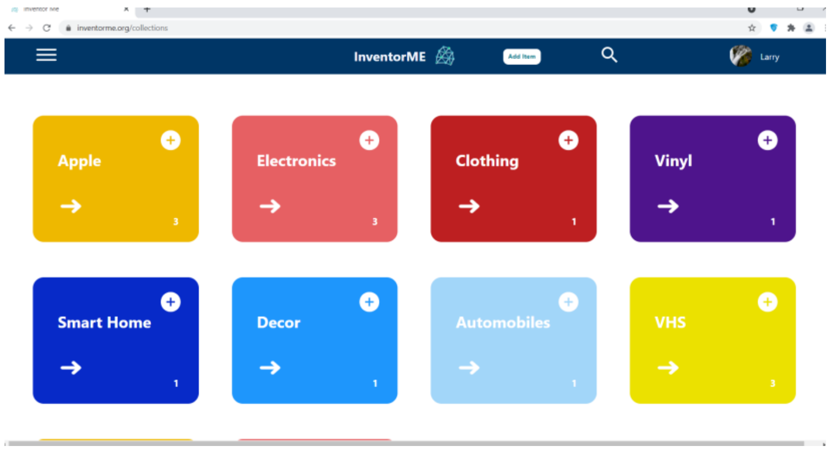
Item Folder*: Favorites

Item Receipt

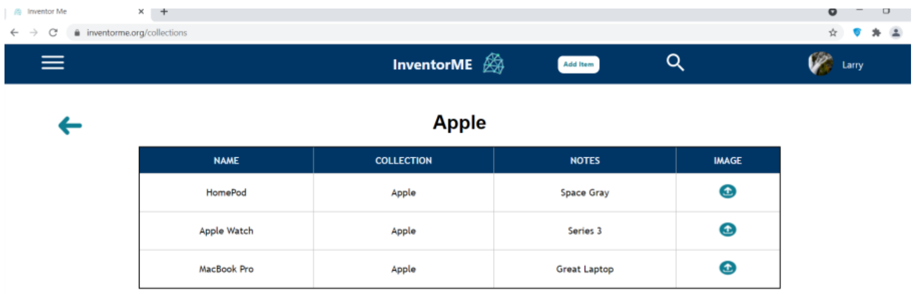
Item Manual Url

Online Url

An example of the collections page:



This is after clicking on the white arrow for the collections page:



Apple

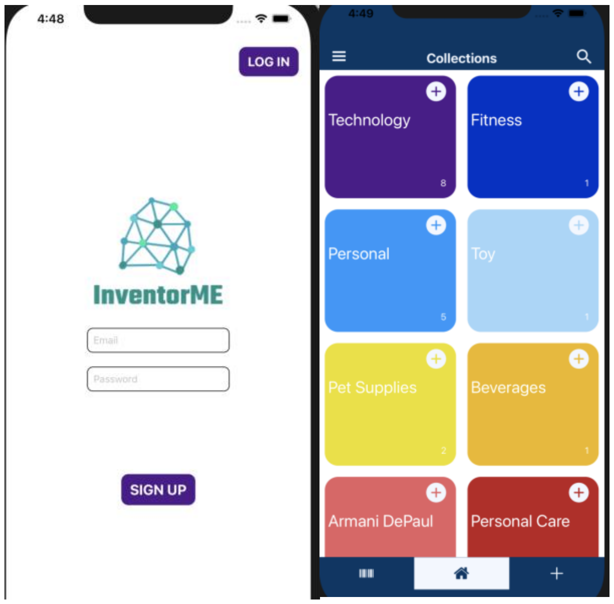
NAME	COLLECTION	NOTES	IMAGE
HomePod	Apple	Space Gray	
Apple Watch	Apple	Series 3	
MacBook Pro	Apple	Great Laptop	

There was a bit of a learning curve when implementing the web app. Some lessons we learned were getting the pages to scale properly, working with different coding styles and personalities, and making everything cohesive across the board. For future work with the web app, we are wanting to make the http requests much faster, get everything to scale better, and allow the user to create custom fields.

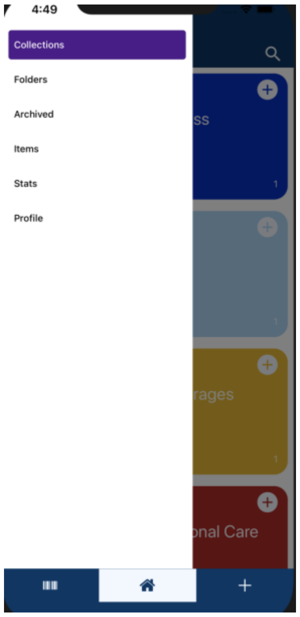
Next, for the mobile application, the web app team applied a similar structure as the web app by utilizing a microservice architecture. In addition to the navigation bar that was reusable across multiple views of the application, Esteban Duran created the Collections using a customized “BlockFolder” component which allowed for reusable code and was also used for the item list view. Anthony Lopez-Guerra created the “ProgressBar” component which allowed for multiple progress bars to be created to track statistical data in the Stats view. The technologies used for the mobile group was React-Native which is like React but allowed for cross platform code between iOS and Android utilizing a dependency called Expo. For the backend we sent requests to a MySQL database that was hosted on AWS and allowed for synchronization across all our applications. The backend will be dissected in greater length, subsequently in the report.

The interface of the mobile applications begins in the Collections view unless a user was to not be logged in. The login page prompts the user to login using their account details or create an account. After a successful login, each view is contained in the Navigation bar to show the user which view they are currently on and to allow easy navigation to other parts of the application. If a user wanted to quickly add an item to the specific collection, they could simply click on the plus icon within the collections component and add an item to that collection. One feature which was exclusively available for the iOS and Android platforms was the Scanner functionality which allowed users to scan an item’s barcode to add said item to a collection or folder. Once scanned, the barcode cross referenced an item database based on the UPC number. If a barcode was not found within the database users could manually enter the data, otherwise the item’s data is automatically populated.

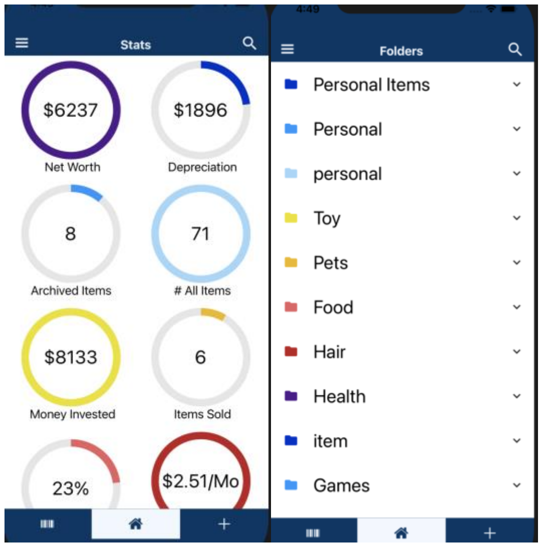
Main mobile interface dependent on sign in state, navigation bar picture on the right:



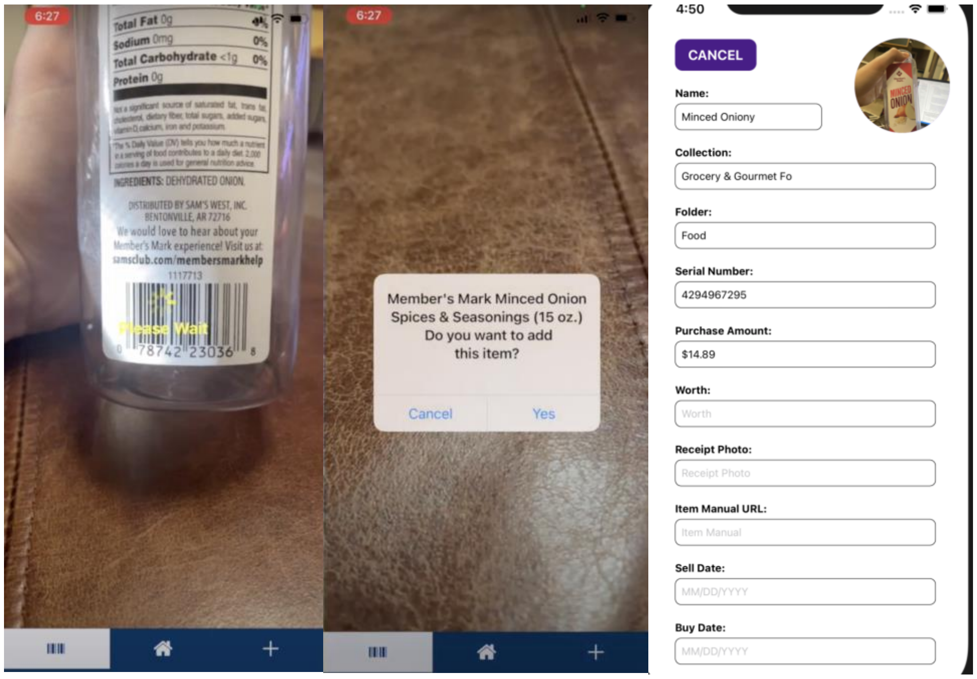
Hamburger nav bar with routes to other views:



The Stats and Folders views:



Scanner view, a successful scan, and EditItems view:



Member's Mark Minced Onion Spices & Seasonings (15 oz.) Do you want to add this item?

Name: Minced Oniony

Collection: Grocery & Gourmet Fo

Folder: Food

Serial Number: 4294967295

Purchase Amount: $14.89

Worth: Worth

Receipt Photo: Receipt Photo

Item Manual URL: Item Manual

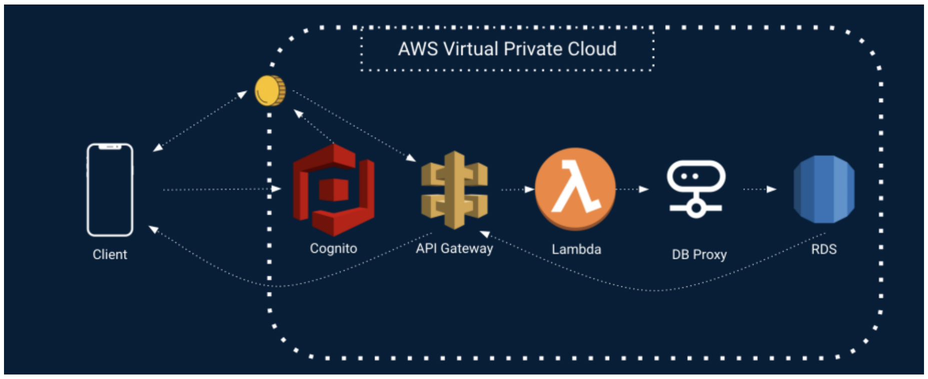
Sell Date: MM/DD/YYYY

Buy Date: MM/DD/YYYY

For the mobile app, some lessons we learned were getting different components to work together, working with different coding styles and personalities, making everything cohesive and there is a lot of work that goes into making a design that is both functional and easy to use. For future work, we want to be able to upload files and documents, and pictures without using a camera. We also want to allow the user to create custom fields. Finally adding the ability for the mobile app to scale better to larger devices, such as iPads.

The backend architecture was built using Amazon Web Services (AWS) tools as well as MySQL for the database. When creating an account, that account is fully managed in AWS Cognito. Cognito is a service that is safe and relatively easy to integrate into the app. That Cognito user then has a unique rotating token that gives them access to the two REST APIs we build for data and image I/O. The REST API talks to a lambda function that determines what the sent/requested data needs to do as well as where it needs to go. Then the lambda function accesses the S3 bucket for images or the database through a proxy, both inaccessible outside of the AWS virtual private cloud, which is like a secure bubble that keeps that data safe from those outside said bubble. This serverless design meant no server management and allowed our backend to act like a server but with way more efficiency and was way easier to scale/maintain. This approach was also much more cost effective and better for the environment because there are no servers running 24/7.

The architecture for the AWS VPC interacting with the client.



For the backend, there was a steep learning curve utilizing the AWS frameworks along with many of the services provided. Some lessons we learned were getting the token to authenticate the user for the Cognito user pool. APIs for data I/O and their respective Lambda functions also were needed in order to fully implement the CRUD functionality. For future work, we want to encrypt data in the VPC so that even we the admins, for privacy reasons, cannot see the users' data and images.

4.3 Risks

Risk	**Risk Reduction**
Confidential Information Leaks	By using an Amazon Virtual Private Cloud, the data is only accessible through the APIs and the AWS private two factor login. This will make it difficult for anyone to access the data, that shouldn't be.
Password Retrieval	With AWS Cognito, they handle all password encryption and account data handling. The email also must be confirmed before an account is usable. For a password to be reset, the confirmed email must be used as well.
Data Loss	The database is constantly being backed up, allowing for data redundancy. It is also scaled by AWS so if more power is needed for the database, then more power is given automatically.
Server Shutdown	Utilizing Amazon Web Services and serverless architecture, there really is not much of a chance for a server shutdown. That is unless AWS is down, which is extraordinarily rare, and if that happens there will be bigger fish to fry, as the saying goes.

4.4 Tasks

1. Understand the problem/objective

2. Implement database tables

3. Create sign in page (web/mobile)

4. Build create account page

5. Create hamburger menu, route to pages (Collections, Folder, Archive, Stats)

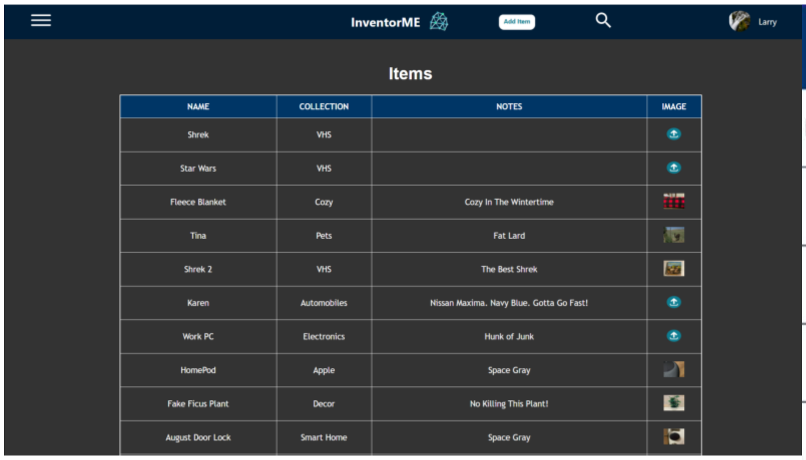
6. Create profile page

7. Implement the http requests for the database (GET, POST, PUT, DELETE)

8. Create home page (web/mobile)

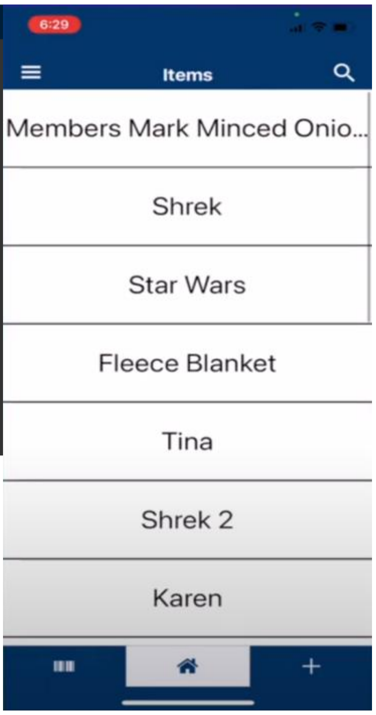
9. Implement sign-in/sign-out functionality (routes to home page/stores a session key)

10. Implement display cards of items in the database



Items

NAME	COLLECTION	NOTES	IMAGE
Shrek	VHS		
Star Wars	VHS		
Fleece Blanket	Cozy	Cozy In The Wintertime	
Tina	Pets	Fat Lard	
Shrek 2	VHS	The Best Shrek	
Karen	Automobiles	Nissan Maxima. Navy Blue. Gotta Go Fast!	
Work PC	Electronics	Hunk of Junk	
HomePod	Apple	Space Gray	
Fake Ficus Plant	Decor	No Killing This Plant!	
August Door Lock	Smart Home	Space Gray	



Members Mark Minced Onio...

Shrek

Star Wars

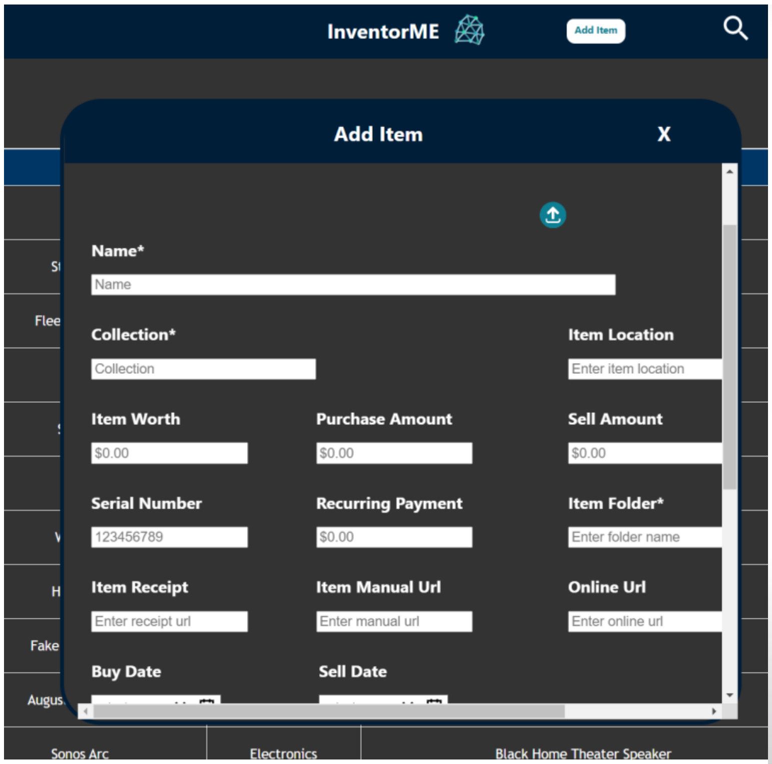
Fleece Blanket

Tina

Shrek 2

Karen

11. Create form page for adding/modifying an entry (web/mobile)



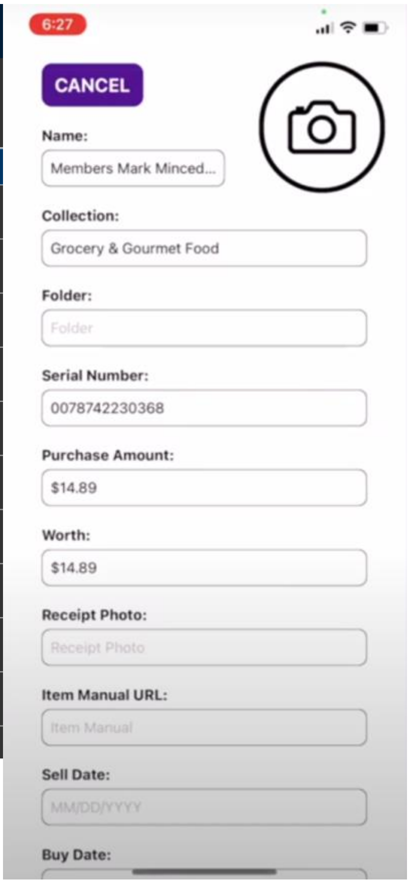
Add Item X

Name*
Name

Collection* | Item Location
Collection | Enter item location

Item Worth	Purchase Amount	Sell Amount
$0.00	$0.00	$0.00
Serial Number	Recurring Payment	Item Folder*
123456789	$0.00	Enter folder name
Item Receipt	Item Manual Url	Online Url
Enter receipt url	Enter manual url	Enter online url
Buy Date	Sell Date	

Sonos Arc | Electronics | Black Home Theater Speaker



Name:
Members Mark Minced...

Collection:
Grocery & Gourmet Food

Folder:
Folder

Serial Number:
0078742230368

Purchase Amount:
$14.89

Worth:
$14.89

Receipt Photo:
Receipt Photo

Item Manual URL:
Item Manual

Sell Date:
MM/DD/YYYY

Buy Date:

12. Implement form functionality (submits to database/editing an entry populates the form)

13. Implement notifications/toast messages (web/mobile)

14. Implement Folders and Archive page

15. Implement Collections and Stats page

16. Implement archive and delete functionality for the entries

17.. Implement search functionality by title, tags, or phrases

18. Implement picture upload / scan functionality

19. Test database

20. Test sign in, search, hamburger menu pages, form interactions

21. Deploy the apps

22. Document any bugs

4.5 Schedule

Tasks	**Dates**
1. Understand the problem/objective	01/11-01/12
2. Implement database tables	01/13-01/25
3. Create sign in page (web/mobile)	
4. Build create account page	
5. Create hamburger menu, route to pages (Collections, Folder, Archive, Stats)	01/26-02/08
6. Create profile page	
7. Implement the http requests for the database (GET, POST, PUT, DELETE)	
8. Create home page (web/mobile)	02/09-02/16
9. Implement sign-in/sign-out functionality (routes to home page/stores a session key)	

10. Implement display cards of items in the database 11. Create form page for adding/modifying an entry (web/mobile) 12. Implement form functionality (submits to database/editing an entry populates the form) 13. Implement notifications/toast messages (web/mobile) 14. Implement Folders and Archive page 15. Implement Collections and Stats page	02/17-02/14 02/15-02/22 02/23-03/13 03/14-03/28

16. Implement archive and delete functionality for the entries 17. Implement search functionality by title, tags, or phrases 18. Implement picture upload / scan functionality	03/29-04/05
19. Test database 20. Test sign in, search, hamburger menu pages, form interactions	04/06-04/16
21. Deploy the apps 22. Document any bugs	04/17-04/30

4.6 Deliverables

- Deployed website:
  - This is the home of the deployed web application.
- Database, Image, and Authentication handling in AWS:
  - The VPC is where the serverless backend it run. This include the APIs, the functions that run said APIs, and the database/S3 bucket where all the data is stored.
- React code for the web application:
  - This is the code that runs the web application.
- React native code for the iOS and Android application:
  - This is the code that runs both the iOS and Android applications.
- GitHub Organization:
  - This is our GitHub organization where all the code for the backend, mobile, and web app resides including schemas for color and databases.

- Trello:
  - This is a link to where we managed our tasks, set goals, set who was to do each task, as well as mark them complete when finished. This all has a breakdown for the future work we would like to complete
- Poster:
  - This is a link to the poster which is an overview of our project: a macro view of what our goals were and what we completed.
- Final Report:
  - This report, what you are reading now, is a complete documentation and, well, report on what our aspirations were, and what we completed.
- Final Presentation:
  - This presentation was a quick review and demonstration of what our team accomplished, as well as great capstone to the project.
- Project Website:
  - The project website is a quick public place for people to go to get clued in to as what the project was, as well as who worked on it.

5.0 Key Personnel

Floyd Brown - Brown is a senior Computer Science major, within the Computer Science and Computer Engineering Department at the University of Arkansas. Several of his skills include an understanding of computer networks, programming languages such as python, java, C++, and a decent understanding of data structures and algorithms. Brown also has experience using specialized applications, such as network sniffers (Wireshark). In general, Brown's skill set is centered around computer networks and basic web development. Brown also has a wide variety of interests, such as hunting, motorcycling, and cooking. He also has a deep affection for coffee. He has completed a wide variety of classes at the University of Arkansas, including: Programming Paradigms, Software Engineering, Algorithms, Formal Languages and Computability, Computer Networks, Cryptography, Artificial Intelligence, and Computer Organization. Brown also interned for the United States Department of Defense in the summer of 2019. He primarily worked on the web application for InventorME, which entailed using tools such as JavaScript (React), HTML, and CSS. He developed several major components using these tools to complete InventorME.

Esteban Duran - Duran is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed: Programming Paradigms, Mobile Programming, Algorithms and Database Management. Duran is a current intern in J.B. Hunt working mainly in mobile development. He is working on new features for the Drive Mobile Application. This Internship has provided him with knowledge in mobile development mainly working with React Native. He was responsible for developing the mobile application for the project with the use of react Native.

Anthony Lopez-Guerra - Lopez-Guerra is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed: Programming Paradigms, Artificial Intelligence, Cryptography, Algorithms, and Formal Languages. This is his second bachelor's degree and has an existing bachelor's in Mechanical Engineering, also acquired at the University of Arkansas. He has prior experience in Project Management and has held previous leadership positions through the Society of Hispanic Professional Engineers. As a computer science major, he has thoroughly enjoyed implementing web apps and mobile applications. He spends his free time taking Udemy courses and self-learning. Over the summer of 2020, he implemented a Ruby on Rails web app which was in the format of a secured blog and launched it on Heroku. He was responsible for assisting in the implementation of the iOS app. He is currently interning for J.B. Hunt, and has learned valuable skills which have assisted him during the semester with his part of the project.

Jaden Martin - Martin is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Paradigms, Software Engineering and Database management courses. Martin currently has an internship with J.B. Hunt, which he has been there for over two years. Throughout his time there he has learned frontend using Angular that connects to a java spring-boot backend. He also worked for Walmart as an intern for two months working with a java backend and database. He was responsible for developing the web application using React.

Luke Miller - Miller is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Artificial Intelligence, Capstone I/II, Mobile Programming, Database Systems, Programming Paradigms, and Software Engineering. Miller has worked with many programs outside of University, such as a development class for Raspberry Pi's, and many extracurricular projects like personal websites and committees aimed towards bringing technology into the classroom of rural high schools. He was responsible for the backend infrastructure needed for the applications and participated in the development of both the mobile and web applications.

Benjamin Thedford - Thedford is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Paradigms, Software Engineering, Database Management, Mobile Programming, and Capstone I/II. He was responsible for the development of the mobile application. Specifically, the creation of the profile pages, edit profile page, styling, and validation.

Chase Zimmerman - Zimmerman is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed (or is enrolled in) Programming Paradigms, Software Engineering, Computer Networks, Computer Graphics, Database Management, Artificial Intelligence, Algorithms, and

Computer Organization. He was responsible for working on and developing the mobile application. The mobile application required the use of HTML, CSS, and JavaScript.

6.0 Facilities and Equipment

Facilities:

- Because of Covid, we did not have much choice when it came to the facility that we worked in. Most of us worked from home, separately. Though, we still plan to meet up to celebrate our hard work these last two semesters, after everyone is fully vaccinated of course.

Equipment:

- We all used our own separate laptops/desktops that we have at home.
- We used GitHub to share code with each other and for version control.

7.0 References

[1] Sortly, https://www.sortly.com/

[2] Android Materials Design Guidelines, https://developer.android.com/design/

[3] Apple Human Interface Design Guidelines, https://developer.apple.com/design/human-interface-guidelines/

[4] Law Insider, Confidential Information Definition, https://www.lawinsider.com/clause/definition-of-confidential-information/_1

[5] Law Insider, Proprietary Information Definition, https://www.lawinsider.com/dictionary/definition-proprietary-information/

[6] Adobe Web Page Design Comprehensive Guide, https://xd.adobe.com/ideas/principles/web-design/web-page-design/