We are designing and implementing a project that tracks the inventory for a small business who distributes their nutrition bars to local stores.

Food Inventory Group – Report 1

Summer 2018 (June 8, 2018)

Joshua Vasil, Tanaja Lloyd, Cofi Tiggs, Edward Cochran, Stephen Elmore

***Small Business Food Inventory Project***

***Semester:*** Summer 2018

***Group Members***: Joshua Vasil, Cofi Tiggs, Tanaja Lloyd, Edward Cochran, Stephen Elmore

***Date:*** June 8, 2018

***Team and Project Description:*** For our project ideas, we brainstormed different categories that would fit an appropriate scope for the time given to complete the necessary work. Ultimately we decided to create a project for a hypothetical private business that sells food through local grocery stores and distributes their product through this network. Our code and database would have to track inventory throughout the multiple partners’ locations and record the stock of our product, so our supply chain runs smoothly and our product is always available at all hypothetical stores. When the database indicates that a store is running low on our product, we would send a new shipment. For this initial report, we brainstormed that our company would sell nutrition bars, and is partnered with local health food stores to distribute this product.

***Technical Outline:*** At the time of this initial report, we have hypothesized that this project will require the use of Java and a MySQL database to effectively complete the goals required during the engineering process. The database would track the inventory of our product at our local partners.

***Team Organization and Roles:*** We have consulted between each other on the best way to organize our team, and so far, have decided to run it democratically, so all ideas must have a consensus before being acted upon. One person is designated as the team “coordinator” to interact with the Instructor in terms of turning in documents and asking questions.

***Table***

***\****within our group, everyone’s role is dynamic and we expect people will take part in a variety of tasks throughout the project, despite what they may be listed as in the initial report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Food Inventory Group* | | | | |
| Joshua Vasil | Tanaja Lloyd | Cofi Tiggs | Edward Cochran | Stephen Elmore |
| Role: Team Coordinator, MySQL Database Management, Java Programmer | Role: Java Programmer, MySQL Database Management, Project Designer | Role: Project Tester and Debugger, Java Programmer, Project Designer | Role: Java Programmer, Report Writer, Project Designer | Role: Project Design, Database Management, Report Writer |

*Individual Resumes*

***Joshua Vasil***

My name is Joshua, and I am a Bachelors student at GSU pursuing a BS in Computer Science. This is my junior year and I specialize in Java and Python development, focusing on these languages in my classes and believe I am proficient in executing tasks using these languages. Some classes I have taken include Data Structures, System Level Programming, Computer Org, and others related to this major.

I have taken part in several projects to further my education, mostly using Java and Python. One included tracking a satellite using Python and displaying it on a map as it orbited the Earth. Another, using Java, was a text based game simulator that allowed you to play simplified versions of Blackjack or Poker. Listed below are skills I have proficiency in.

* Languages: Java, Python, JavaScript, C
* Skills and Programs: Excel, Word, Matlab, MySQL Database Management
* OS: Linux (Ubuntu, Debian), MacOS, Windows

I also have some experience in Web Development, taking part in the design and programming of a local business who needed their page redesigned.

***Tanaja Lloyd***

|  |  |
| --- | --- |
| **Education** | **Computer Science**  Georgia State, Atlanta, GA   * Data Structures * JAVA Programming * Computer Architecture * Design and Analysis * Web Programming (HTML, JavaScript, PHP), MySQL |
| **Technical Skills** | Skilled at JAVA Programming: data structures, analysis.  Web Development: Notepad++, Visual Studios, HTML, PHP, CSS, XML, MySQL, and JavaScript.  Microsoft Suite: Word, Excel, PowerPoint.  Server: Ubuntu, MAMP, SSH tool through Terminal or Putty. |
| **Experience** | * Developed a blog website that restricted access for different levels of users using information from SQL database: my part was to create, alter, compare input against, and display the database contents onto the website pages. * Developed an app that enabled students to register for classes. |

***Cofi Tiggs***

Hello, my name is Cofi. I’ve almost graduated, so, naturally, I’ve taken a lot of CS courses, including, but not limited to, the following:

* Design/Analysis of Algorithms
* Data Structures
* Computer Org. (with Mussa)

Personally, I’d like to describe myself as an average coder. This will be my first non-hackathon project that I’ve finished with a team. I prefer to use Java as its my most comfortable language, but I can also learn another for convenience of the project. So, as far as my role on the team, I would be flexible in my assignment. I don’t really prefer to code because I hate the pressure (sometimes it makes me panic really bad) and my time this summer isn’t consistent as it possibly should be. But, I’ll have to make time somewhere, so I guess that’s a moot point. Also, I am big on grammar, spelling, etc., so I can type up reports that we need if necessary.

Skills:

* Proficient at Java
* Web: HTML, CSS
* Office Suite: Word, Excel, PowerPoint

Experience:

* Developed a Sudoku web app (without reusing logic code)
* Developed app to simulate a vending-machine

***Edward Cochran***

|  |  |
| --- | --- |
| **Education** | **Computer Science**  Georgia State, Atlanta, GA   * Data Structures * JAVA * Computer Architecture * Design and Analysis * Computer Org |
| **Technical Skills** | Proficient in Java  Microsoft Suite: Word, Excel, PowerPoint.  Server: Ubuntu and SSH tool through Terminal or Putty. |
| **Experience** | * Created a vending machine stocking program that read which items were most popular, checked sales per item and let me know how many of each item are left in the machine |

***Stephen Elmore***

**Education**

* Georgia State University (Current Enrollment): Bachelors in Computer Science
  + JAVA Programming
  + Assembly Language
* Ola High School (2013)

**Skills**

* Experience in JAVA
* Basic knowledge in Assembly Language with Visual Studio, UNIX, and C#

***Small Business Food Inventory Project – Report 2***

***Requirement Engineering***

***Semester:*** Summer 2018

***Group Members***: Joshua Vasil, Cofi Tiggs, Tanaja Lloyd, Edward Cochran, Stephen Elmore

***Date:*** June 15, 2018

***Core Tasks and Milestones:*** To complete our project within the timeframe given, we must stick to a schedule that lays out core tasks to be completed each week and pace our work. At this point in the process, we are two weeks in and have completed a general outline for our project and laid out the requirements engineering part of the process that sets a schedule, tasks to be completed, and collaboration system for all team members.

* The software and hardware requirements for our system will be minimal, as it must run on a basic computer within the means of the company we are creating the project for. In terms of software, you must be able to run a Java program and access a MySQL database with an internet connection. Our hardware is also minimal, a computer with a modern CPU will be able to run the software.

***Challenges and Risks:***

* Our most serious challenge developing the product on this timeframe is going to be the design and implementation phase, where we must create a functioning system that can track inventory and manage the communications between a database and its nodes. This will take a rigorous application of our programming knowledge to create a finished product that meets all requirements and expectations.
* The best way to minimize risk for this type of project is constant communication between the team to ensure all work is being done on time and per schedule, so we can account and adapt to any problems that may arise throughout the process without compromising the overall ability to meet our deadline.
* The objective overall for our product is to streamline and make more efficient the system in which our business interacts with the stores it distributes product to.

***Scheduling:***

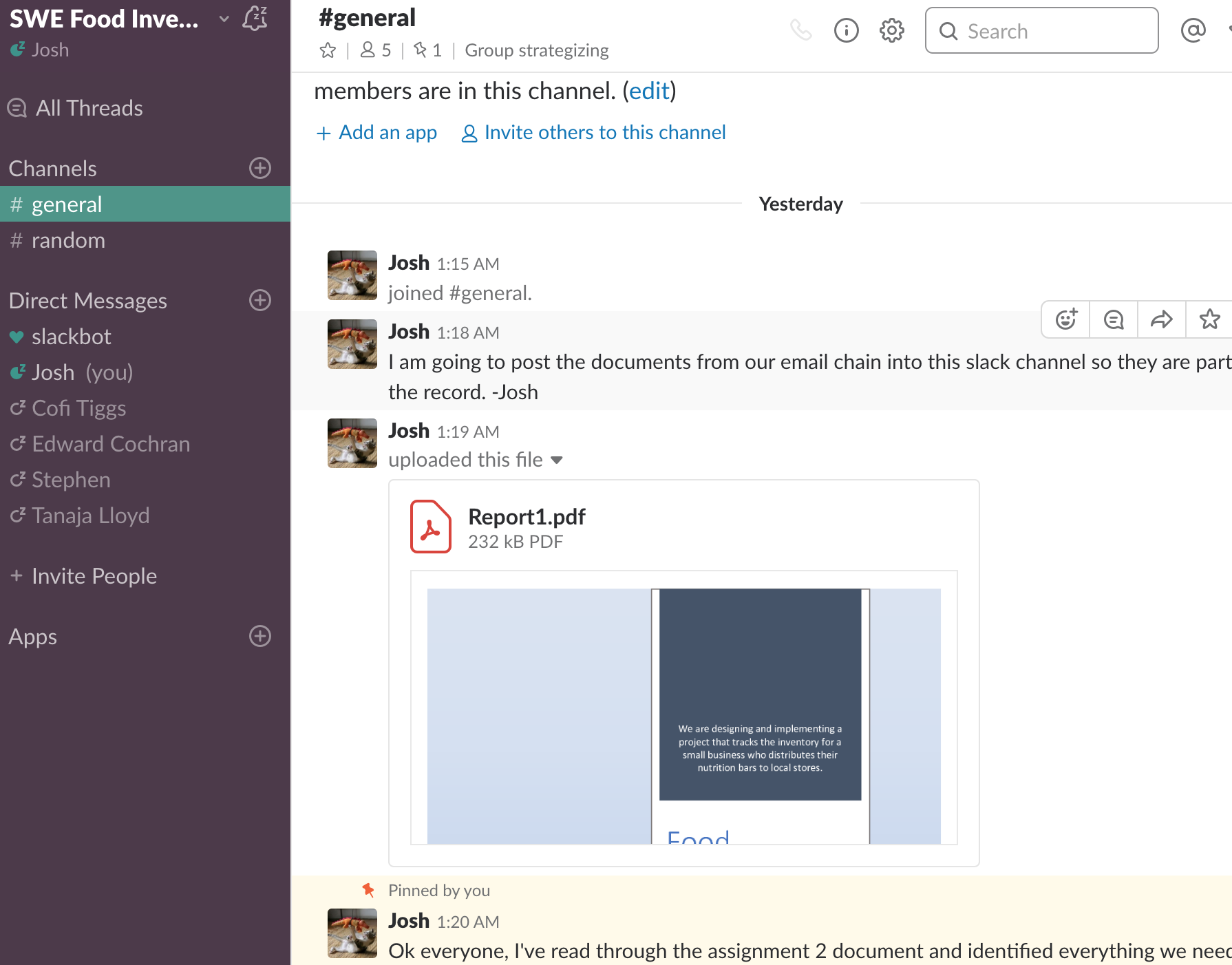
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Food Inventory Group* | | | | |
| Task | Effort | Duration | Dependencies | Assigned To |
| Database Design | Creating a visual, identifying tech requirements | 1 person hours, | none | Stephen |
| Database Creation | Setting up MySQL database | 4 person hours, | Design completed | Josh |
| Server Set up | Identifying communication method between database | 1 person hours, | none | Tanaja |
| MySQL Programming | Programming database connections | 4 person hours, | Database created | Josh and Edward |
| Connecting DB Nodes (stores with database) | Integrating stores with our database | 3 person hours, | Database functional | Josh |
| Java Communicating with Database | Using java to speak with the database | 3 person hours, | Database functional | Cofi and Tanaja |
| UI Design | Sketching out and designing user interface | 5 person hours, | none | Stephen |
| UI Programming | Programming UI in JAVA | 3 person hours, | none | Cofi and Edward |
| Final Debugging | Testing complete system | 2 person hours, | none | Stephen and Tanaja |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

***Collaboration (Appendix)***

***Github:*** The link to our Github can be found [here.](https://github.com/joshvasil/SWEInventoryProject)

***Slack:*** The link to our Slack logs can be found [here.](https://pastebin.com/ZWYFUMrr)

***Video:*** The link to our video can be found [here.](https://youtu.be/4yYW3TRRr9k)



***Problem Statement:***

* ***Target Customers, Differentiators, and Top Level Objectives:***
* At a high level, our ***top- level objective*** is to create a database with nodes corresponding to stores that carry product that needs to be tracked, and some complimentary code that connects all of this together and shows the end user a comprehensive display of their inventory in all locations.
* The ***target customer*** is a food company that makes nutrition bars and distributes them to local health food stores. They need a way to track their inventory in all these locations.
* The product solves the problem of not having an efficient way to track inventory. Without our product, the food company would have to contact each store individually and have a human tell them how much inventory was in stock.
* There technological use of a database and fulfillment system is a ***differentiator*** between our product and alternatives, such as manually tracking inventory without the use of technology, or going with a competing product, perhaps offered by a third party, and attempting to integrate their alternative product into the inventory network specified by our outline in Report 1.
* This product is compelling and worth developing because it saves time, money, and effort for the end user. It is a simple but effective program that integrates important information into one location allowing the end user to make critical decisions.

***System Requirements:*** Our system will require interaction between our ***UI***, the ***database and its nodes***, and ***the actors (store and company)*** to be functional. Below is our running list of requirements, which is updated as we move through the project.

**1**

* **Requirement ID:** Requirement 1 – **Request Inventory**
* **Requirement Description:** This requirement entails that an actor, (a food store), can use our system to make an order of more inventory.
* **Rationale:** This requirement was included as it is a core function of achieving our systems goal, which is managing orders of inventory between supplying company and store.
* **Inputs:** The input would be how much of the product is requested in an order.
* **Outputs:** The output would be the food company receiving an order with Product – Product ID- How much of Product Requested.
* **Persistent Changes:** A persistent change from this requirement is that it will enter a new order into the log of all requests made using our system.
* **Related Requirements and Use Cases:** in Progress
* **Test Cases:** in Progress

**2**

* **Requirement ID:** Requirement 2 – **Fulfill Order**
* **Requirement Description:** This requirement entails that the food company responds to an order request, choosing to fulfill it and send more product to a store using our system.
* **Rationale:** This requirement was included as it is a core function of achieving our systems goal, which is managing orders of inventory between supplying company and store.
* **Inputs:** The input is response the company gives choosing to fulfill an order.
* **Outputs:** The output would be a grocery store receiving a carton or box of new product, because the order was fulfilled.
* **Persistent Changes:** A persistent change from this requirement is that it will enter a new order into the log of all requests made using our system.
* **Related Requirements and Use Cases:** in Progress
* **Test Cases:** in Progress

**3**

* **Requirement ID:** Requirement 3 – **Order Log**
* **Requirement Description:** This requirement entails that a food company can use our system to display a history of all orders received from stores.
* **Rationale:** This requirement fulfills the function of tracking orders and inventory across multiple stores.
* **Inputs:** The input would be each order that is made from a store.
* **Outputs:** The output would be an order being added to the global log of all orders made.
* **Persistent Changes:** A persistent change from this requirement is that it will enter a new order into the log of all requests made using our system.
* **Related Requirements and Use Cases:** in Progress
* **Test Cases:** in Progress

***Use Case:***

* **Summary:** The food company distributes food to stores who sell it and request more inventory.
* **Basic Course of Events:**
  1. The grocery store sells the product and inventory decreases.
  2. The store completes use case *Request More Inventory*
  3. The food company receives this message. (*Send Message)*
  4. They complete use case *Check Inventory at Stores.*
  5. Completion of use case *Distribute Product*
* **Alternative Paths:** In steps 2, the grocery store can alternatively complete use case *Send Message* to convey a more detailed description of current inventory levels rather than generic *Request More Inventory.*
* **Exception Paths:** In step 4, if the food company checks the inventory and does not see it is low, they can alternatively choose to send a message asking for further clarification before proceeding to step 5.
* **Precondition:** The Request More Inventory use case is completed successfully.
* **Postcondition:** The product is distributed to the grocery store from the food company.

Below is a use case diagram displaying the multiple different actors, systems, and tasks at hand that interact together to form our product. As shown, you can see humans and the actual system included in the diagram.

Inventory System

Grocery Store

Food Company

Distribute Product

Send Message

Check Inventory at Stores

Sell Snack Bar

Receive Product

Request More Inventory

***Small Business Food Inventory Project – Report 3***

***Semester:*** Summer 2018

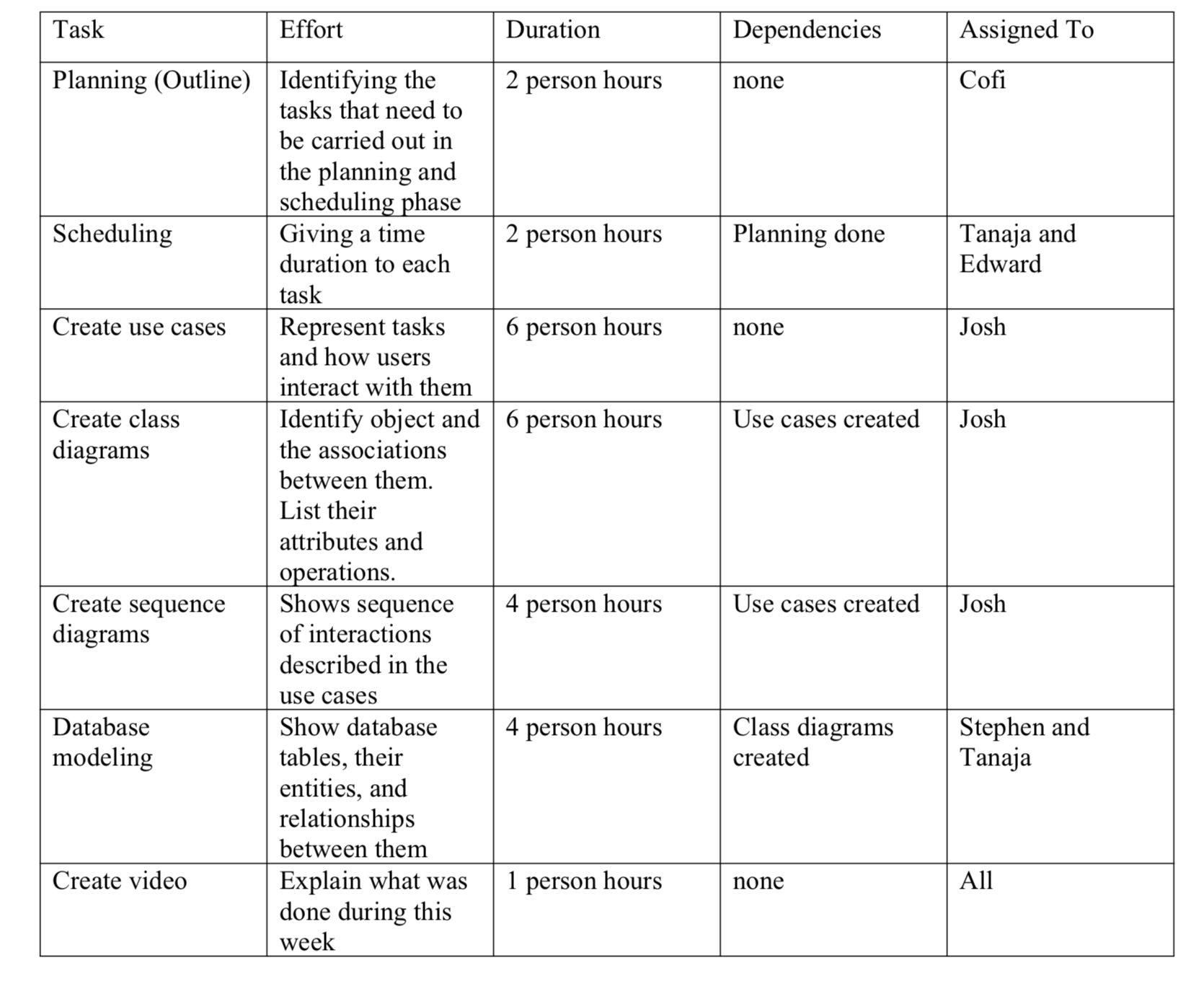
***Group Members***: Joshua Vasil, Cofi Tiggs, Tanaja Lloyd, Edward Cochran, Stephen Elmore

***Date:*** June 22, 2018

***System Modeling***

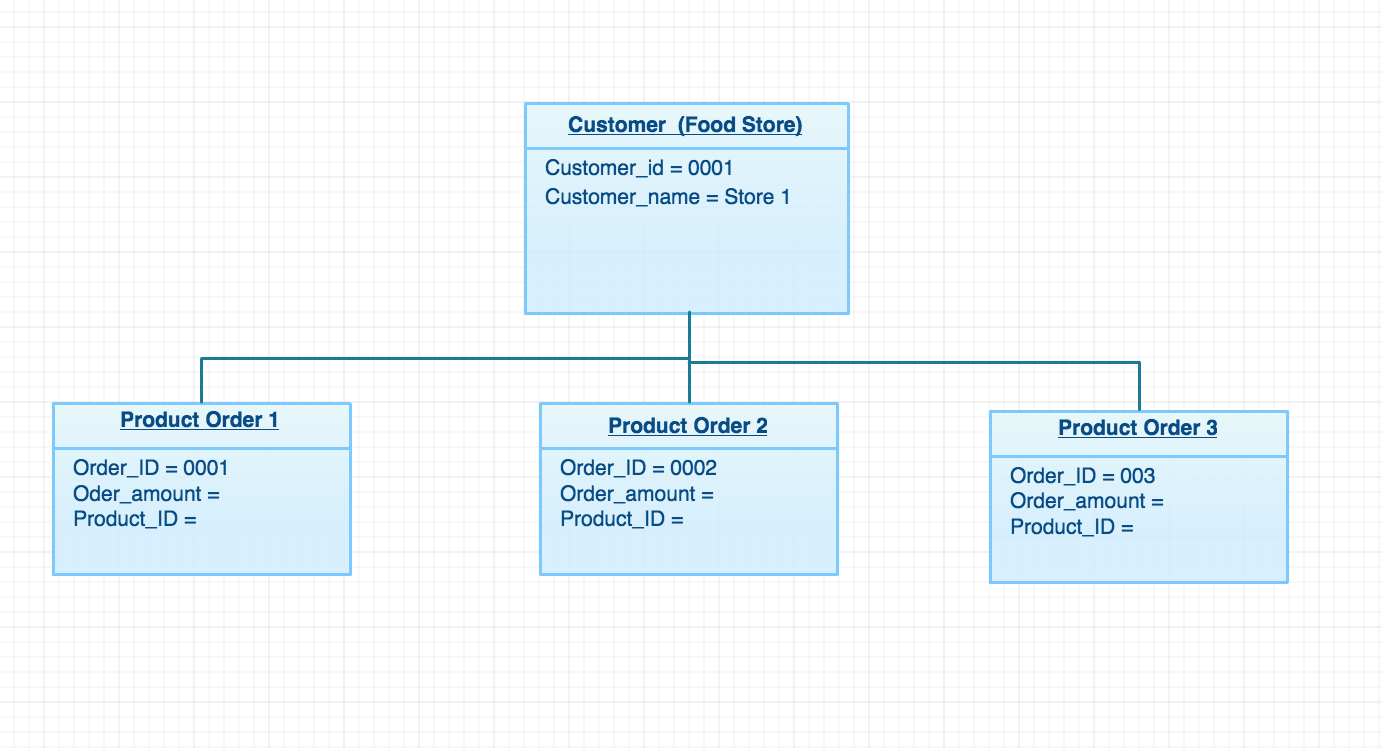
***Outline:***

***Schedule:***

******

***Objects:*** Our project system has several ***objects*** in it that work together as a whole. These interactions include the *company, the customer, and the orders between them*. This is represented in the following object diagram.

*Object Diagram 1*



***Test Cases:***

* 1. **Get Log of Orders**

**Description:** Checks that orders are recorded in a log

**Test Inputs:** Order database initialized with id\_store, id\_carton, id\_parcel tables.

**Expected Results:** A list of orders is returned. Each one with the above attributes.

**Dependencies:** None

**Initialization:** Database is loaded.

**Test Steps:**

1. Request a new order from food supplier
2. Ensure the order is fulfilled
3. Verify that it is recorded in the order log
   1. **Get Product Request**

**Description:** Checks that product can be ordered from supplier

**Test Inputs:** Order database initialized with “order” class.

**Expected Results:** The supplier receives a new request for product.

**Dependencies:** None

**Initialization:** Database nodes are communicating with each other.

**Test Steps:**

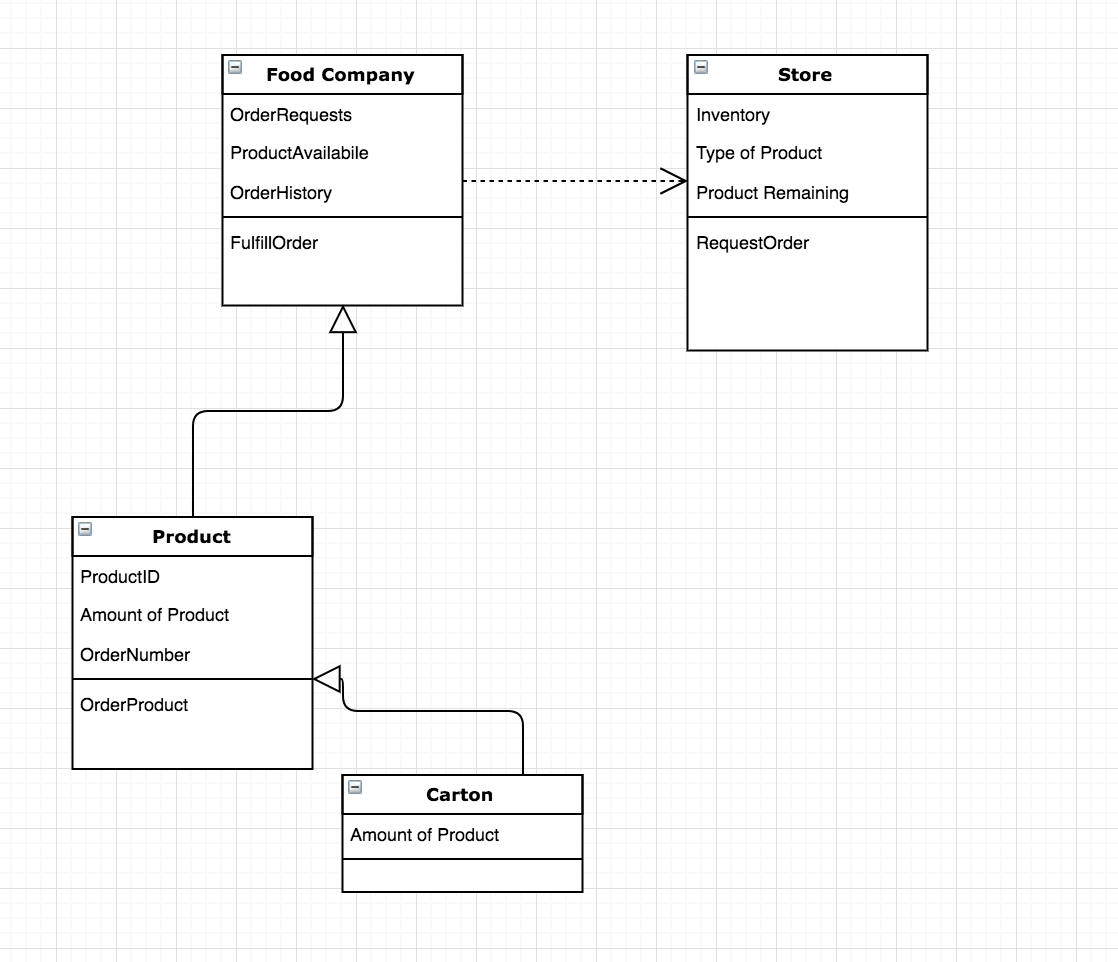
1. Request a new order of product because of low stock
2. Check that supplier receives order request
3. Check that the product request is fulfilled

***Class Diagram:***

Our class diagram shows the relationship between classes and their hierarchy.

* Each class in the diagram has ***names*** (e.g. Product), ***operations*** (e.g. OrderProduct), and ***attributes*** (e.g. ProductID) assigned to it.
* The behavior of each class is represented by its ***operations.***
* The structure of each class is determined by its *attributes*, each one required before a class can be complete and its operation executed.

*Class Diagram 1*



***Object Associations:***

* The “Product” object has ***unary association*** with the object Food Company, and Store.
* The “Carton” object is unary associated with the “Product” object.
* The two objects “Food Company” and “Store” have *unary association* with each other.

***Multiplicity:***

* For our system, the multiplicity of the product being requested in *Class Diagram 1* is **1..\***, or, **zero or more instances**, because for a request to be valid at least one product must be requested.

***Attributes:***

* The “Store” object has attributes *Inventory, Type of Product, and Product Remaining*, and then action *RequestOrder*.
* The “Food Company” object has attributes OrderRequests, ProductAvailabile, and OrderHistory, and then action FulfillOrder.
* The “Product” object has attributes *ProductID, Amount of Product, and OrderNumber,* and then the action “*OrderProduct*”.
* The “Carton” object has an attribute called “*Amount of Product*”.

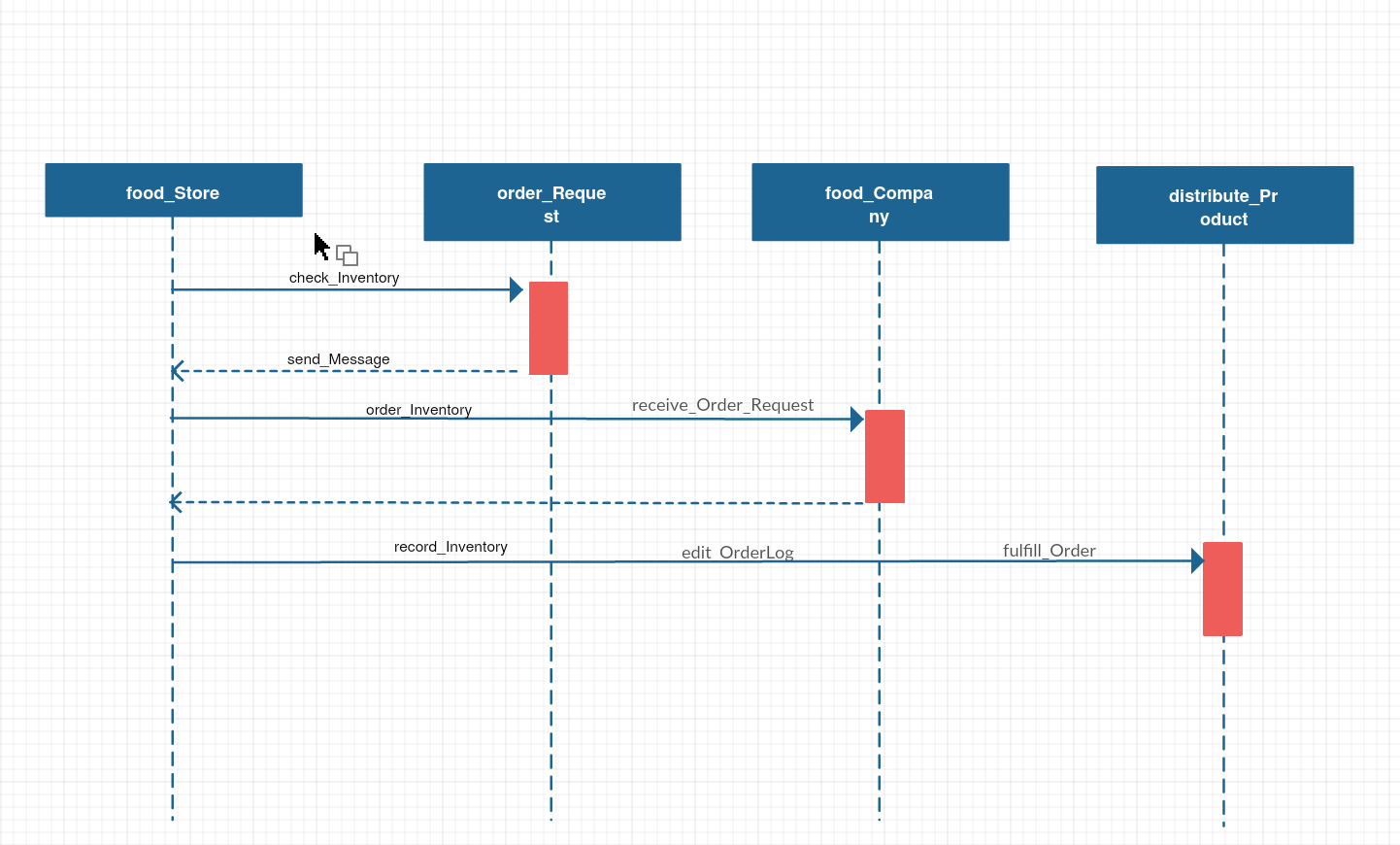
***Operations:***

* The objects have three operations total, consisting of *RequestOrder, FulfillOrder,* and *OrderProduct.*
* These operations encompass the steps required to complete the process of a store with low inventory making an order for more product, and then the supplier fulfilling that order.

***Sequence Diagram:***

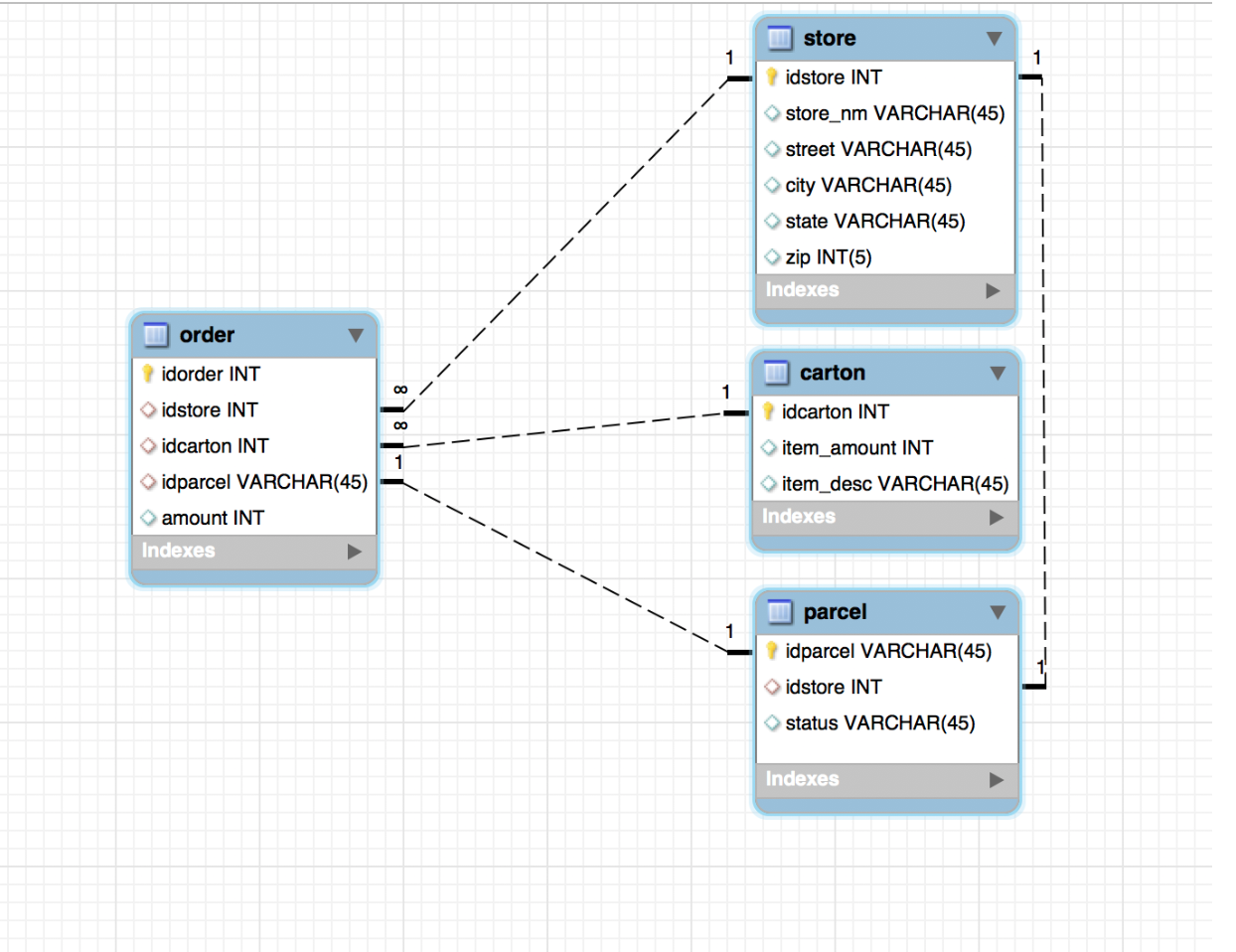
* Our sequence diagram models the interaction between the store **(actor)** and its product requests from the supplier **(objects).**
* The purpose of this diagram is to show the *interactions* that happen during our use case scenario presented above.
* It also helps identify extra objects and new responsibilities that each class in the system may have.
* In the diagram below, the identifiers food\_Store, order\_Request, food\_Company, and distribute\_Product are ***Instances***.
* The identifiers check\_Inventory, send\_Message, order\_Inventory, receive\_Order\_Request, record\_Inventory, edit\_orderLog, and fulfill\_Order are all ***Lifelines*** and ***Messages***.

*Sequence Diagram 1*

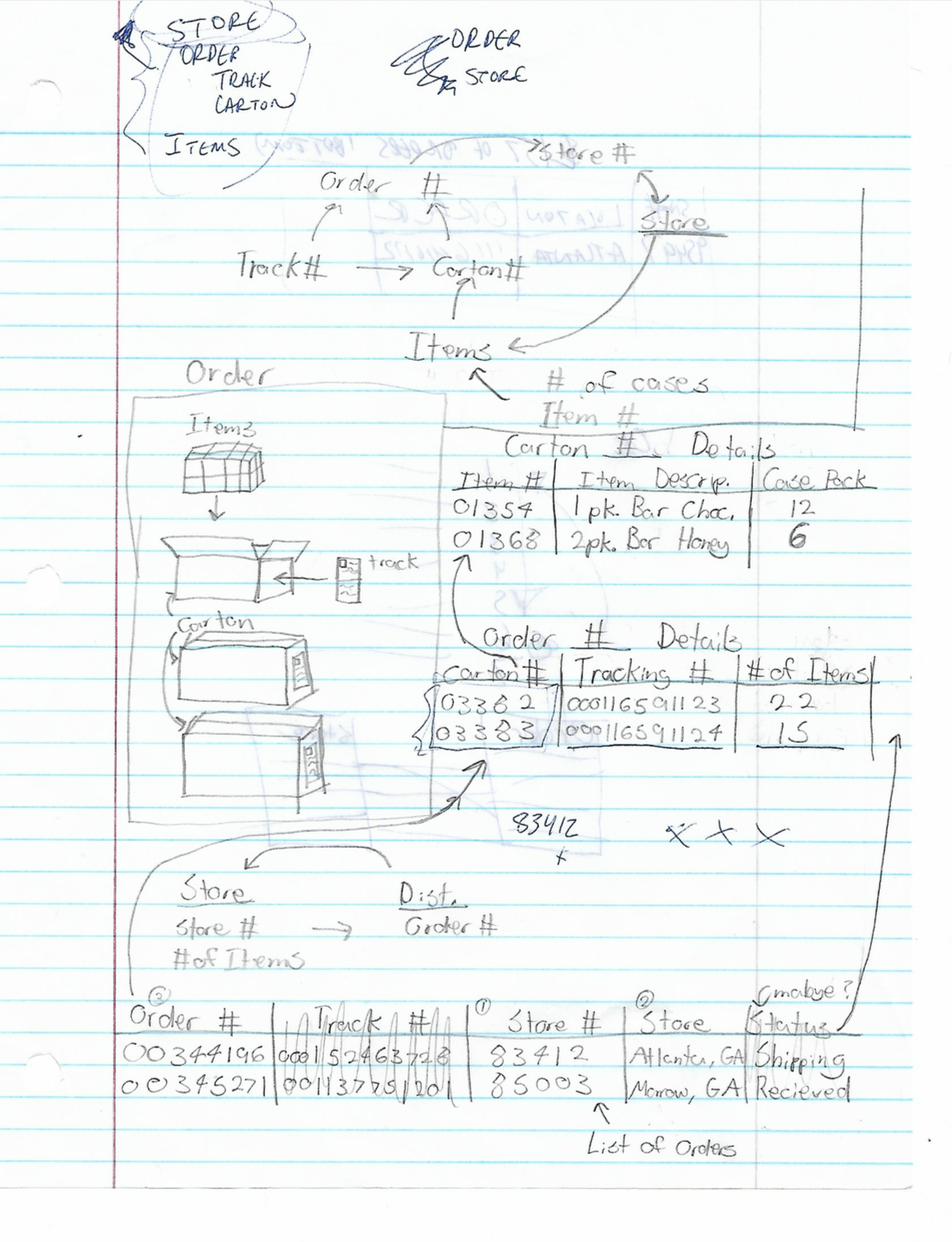
**

***Database:***

* We are using a MySQL database, with four ***tables***—order, store, carton, and parcel.
* Order will have a ***primary key*** (idorder), amount, and 3 ***foreign keys***, idstore, idcarton, and id parcel, which pull from their respective tables.
* Store will have a *primary key* (id store), and 5 other columns that identify the store name and location.
* Carton will have one *primary* *key* (idcarton) item\_amount which tells how many of the items are in the carton, and item\_desc which is a description of the item.
* Parcel will have one *primary* *key* (idparcel), status, and a *foreign key*, idstore.

**

*DB Sketch 1*



***Collaboration (Appendix) Report 3 Update***

***Github:*** The link to our Github can be found [here.](https://github.com/joshvasil/SWEInventoryProject)

***Slack:*** The link to our Slack logs can be found [here.](https://pastebin.com/ZWYFUMrr)

***Video:*** The link to our video can be found [here.](https://www.youtube.com/watch?v=ft4PTUbP1LE&frags=pl%2Cwn)