Saint petersburg college

Gundam auto system

denzel coleman

Contents

[Business Case 3](#_Toc170580133)

[System Development 4](#_Toc170580134)

[System Acquisition 6](#_Toc170580135)

[Project Plan (Gantt Chart) 7](#_Toc170580136)

[Feasibility Analysis 7](#_Toc170580137)

[ Economic Feasibility Analysis 7](#_Toc170580138)

[ Technical Feasibility Analysis 7](#_Toc170580139)

[ Operational Feasibility Analysis 8](#_Toc170580140)

[ Schedule Feasibility Analysis 8](#_Toc170580141)

[ Legal and Contractual Feasibility Analysis 8](#_Toc170580142)

[ Political Feasibility Analysis 8](#_Toc170580143)

[Requirements Gathering 8](#_Toc170580144)

[ Interviews 8](#_Toc170580145)

[ Observations 9](#_Toc170580146)

[ Documentation 10](#_Toc170580147)

[Context Diagram 12](#_Toc170580148)

[Use Case 13](#_Toc170580149)

[Dialogue Diagram 15](#_Toc170580150)

[Prototypes 16](#_Toc170580151)

[Implementation Plan 19](#_Toc170580152)

[Maintenance Plan 20](#_Toc170580153)

Table of Figures

[Figure 1 - Project Plan Gantt Chart Embedded Link 7](#_Toc170580119)

[Figure 2 - Economic Feasibility Analysis Embedded Link 7](#_Toc170580120)

[Figure 3 - Context Diagram 12](#_Toc170580121)

[Figure 4 - Receiving Use Case Diagram 13](#_Toc170580122)

[Figure 5 - Site Manager Use Case Diagram 13](#_Toc170580123)

[Figure 6 - Counter Person Use Case Diagram 14](#_Toc170580124)

[Figure 7 - Accounting Use Case Diagram 14](#_Toc170580125)

[Figure 8 - Dialogue Diagram 15](#_Toc170580126)

[Figure 9 - Prototype Logon Page 16](#_Toc170580127)

[Figure 10 - Prototype Inventory Page 16](#_Toc170580128)

[Figure 11 - Prototype Order Page 17](#_Toc170580129)

[Figure 12 - Prototype Vendor Page 17](#_Toc170580130)

[Figure 13 - Prototype Receiving Page 18](#_Toc170580131)

[Figure 14 - Prototype Accounting Page 18](#_Toc170580132)

# Business Case

The goal of our business is to manage the City of Tampa’s inventory for their automotive repair needs for their emergency vehicles such as Police, Fire, and Ambulatory units. The process of managing this inventory involves the ordering, stocking, and supplying of automotive parts. Technicians approach our counters at the storage room to order parts with their respective work orders and we provide whatever it is that they need to complete the repairs on the vehicles.

The software that I would like to propose to the company is an Inventory Management System called Gundam Auto System (GAS.) The intention of this software is to help employees keep track of current inventory in real time to minimize errors, meet our client’s demands, implement a better organization of inventory, and improve overall productivity for the company.

The company has been struggling to keep the city as a key client due to a few complications that need to be worked on. Some of those issues are as follows with solutions:

* Inventory shortage and overages:
  + The company has been experiencing difficulties with keeping track of inventory levels, resulting in many shortages. Because of this, we have been delaying the time for our client to receive the needed automotive part to complete their jobs in a timely manner. On the other end of this issue, the company has also had overstocking issues as well which have been costing the company financially due to having automotive parts stocked on the shelves that are not being used, yet still being ordered.
* Manual inventory tracking:
  + Our current process is to have associates manually enter inventory into a spreadsheet. This process can be time-consuming and can lead to discrepancies in our inventory.
  + Inventory is also manually placed on shelves without any sort of bin location/bar code system. Because of this, there is unknown where parts are physically located and can cause delays in our client’s immediate need for these parts.
* Outdated ordering system:
  + Currently, the process is to have an associate manually call-in or email orders to our suppliers based on either what they visually see from empty spots on the shelves, or as the technician needs to complete their work order. This can cause delays in fulfilling the technicians order and proves to be rather ineffective.

The inventory management system can provide solutions to these problems as follows:

* Suggested inventory levels:
  + The GAS will be able to gather data based on usage for specific vehicles and can recommend stocking levels for each individual automotive part to ensure that the shelves are properly stocked along with not over ordering parts that are not needed.
* Automatic order fulfillment:
  + GAS will be able to automatically contact suppliers based on low inventory levels that are set up based on usage data for each unit that the city provides to our company. The system will also be able to set up minimum and maximum stock levels so that parts will always be ordered within an as needed threshold.
* Inventory tracking:
  + GAS will be able to utilize barcodes, shelves, bins, for tracking. Parts can be labeled with barcodes so that they can view current part locations, used to track any history of a sale, or change of part locations as well as provide real-time quantity on hands.

The utilization of GAS within the company will bring many benefits for our associates and our client, the City of Tampa. The system can lead to massive savings by no longer having discrepancies in our inventory levels by not over-ordering and being able to physically locate each part with no hassle.

Stocking parts that are not used as often based on history usage as well as bring forth efficiency with saving on time and manpower by not having our day-to-day processes done manually.

By setting up minimum and maximum stock levels based on this history for each individual part, we can ensure that we will always have the right amount of inventory that we need to properly support our client. Our main priority is to bring excellent customer service to our client and provide efficient results, and I am confident that GAS will be able to provide those results.

# System Development

* Traditional waterfall: A system development process where the progress on the creation of the software “flows” in one direction from phase to phase just like a waterfall would. The flow of this process is as follows: Planning, Analysis, Logical Design, Physical Design, Implementation, and the final step being Maintenance.
* Agile Methodologies: A system development process that utilizes three key principles:

1. A focus on adaptive rather than predictive methodologies.
2. A focus on people rather than roles.
3. A Focus on self-adaptive process.

* eXtreme Programming: An approach to software development that is most recognized by its short cycles and step-by-step planning, emphasized focus on automated tests that are written by two team programmers as well as the involvement of customer input during the development process. There are two parts that are relevant to design specifications are:

1. How planning, analysis, design, and construction are all intertwined and executed as one activity
2. The unique way of how the programming can capture and present a system requirement and design specifications.

* Scrum: The most popular methodology within agile as it uses teams that have dedicated teams with their own roles and rules. Each team is broken down into three roles: the product owner, the development team and the “Scrum Master.” This system development method was also designed for speed.
* Object-Oriented Analysis and Design (OOAD): Also known as the standard for system development, this process combines data and processes (also known as methods) into a single entity called an object.
* Relational Unified Process (RUP): An object-oriented systems development. This development method is based on a repetitive approach to systems development, has four phases:

1. Inception
2. Elaboration
3. Construction
4. Transition

The methods that I recommend being planned on implementing into the development of this software will be the traditional waterfall method. I decided on a traditional waterfall for the sake of keeping things simple and structured in a way that can be planned step by step, that way we can provide proper updates on how the development of this software is coming along to present to our client. This is also software that the city would like to be operational for many years to come.

# System Acquisition

* Information Technology Services: IT service firms that are highly experienced, help companies develop internal information systems as well as develop, host, and run applications for their clients. Some examples of these firms include IBM and Accenture.
* Packaged Software Producers: These are companies that strictly only produce software that is not customizable. Some examples of these providers include big names such as Microsoft, Oracle, and Inuit.
* Enterprise solutions software: Also known as enterprise resource planning systems (ERP) that support business operations and their processes and consist of a series of modules that support functions such as distribution or manufacturing, for example. One of the biggest names in ERP systems is that of SAP AG.
* Cloud computing: Refers to providing a service or application over the internet. This process is convenient for customers that do not want to invest in software or its development as well as its maintenance. Some examples of cloud computing include the ever-popular Google with its Google Drive, Docs, and Sheets applications.
* Open-source software: Software that is freely available to any organization that was developed by a community versus that of another company. Examples of these would be web browsers such as FireFox, operating systems such as Linux and databases systems such as mySQL.
* In-House Development: This is a source that uses an organization’s staff to create any needed systems. This method involves creating a system from scratch and usually leads to high maintenance on a regular basis.

The sources that I will be utilizing in the development of this software will be Cloud computing, packaged software, and open-source software. The utilization of cloud computing will be a great help for uploading invoices, providing a purchase history, and uploading documentation for both our company and clients to be able to view as we so choose.

Packaged software will also be a great help with programs such as Microsoft Office and being able to create process documents that show step by step instructions on how to utilize the new software, through Microsoft Word as well as create presentations on budget spending with our client and vendors.

The most important open-source software that would be utilized in the making of this software would be that of mySQL to be able to manage suppliers, product inventory, purchase orders, and pricing for individual parts.

# Project Plan (Gantt Chart)



Figure 1 - Project Plan Gantt Chart Embedded Link

# Feasibility Analysis

* Economic Feasibility Analysis: The cost of creating an inventory management system software range from a simple level of operation at around $20k and can go as high up to an extremely complex operation ranging $70k+. For this operation to be successful with very little maintenance in the future, we are going to be looking at a medium range that will cost the business:
  + A Tangible benefit of $2000 of for training and new hardware for the associates.
  + A one-time cost of $57,000 for the development of this software.
  + A recurring cost of $5000 during the development of the software that include data storage, new software lease, and software maintenance.



Figure 2 - Economic Feasibility Analysis Embedded Link

* Technical Feasibility Analysis: This software will be able to be developed without any technology or financial constraints. We currently are utilizing Microsoft Windows, so the development of this software should not be an issue at all. There also should not be an issue with a risk factor, since this is a small project with a projected completion date within 100 days of starting.
* Operational Feasibility Analysis: As far as operations goes, our users should not have an issue with utilizing the GAS inventory system. This software will be a quality-of-life change from new to old associates and should be a user-friendly system. Associates will receive the proper training and support needed to learn and use this software so that they can feel comfortable using it to serve our clients and fulfill their needs.
* Schedule Feasibility Analysis: The projected timeframe for this project to be completed is estimated to be 95 days. Even with a couple of days over, I think anywhere between 95 and 115 days is feasible to complete this project. By utilizing the Gantt chart we have created, we should be able to follow that schedule with no issues, and should anything arise, we will be able to adjust as needed and still complete the project within a reasonable timeframe.
* Legal and Contractual Feasibility Analysis: Our company should not have any issues with potential legal ramifications in the creation of our inventory management system. We have identified any legal risks and liabilities that could be associated with the creation of this project such as insurance but have not come across any issues with it specifically. We also will not be violating any contractual agreements with any of our suppliers, partners or stakeholders.
* Political Feasibility Analysis: We have conducted surveys with our clients here in the City of Tampa to ensure that this project would be a great fit for them and have passed that through our company as well. The Mayor of the City acknowledged that this software will save on time and help get emergency vehicles back on the road within a reasonable time to be able to assist the citizens of the City of Tampa.

References:

<https://kodytechnolab.com/blog/cost-for-developing-warehouse-management-software/>

# Requirements Gathering

* Interviews: Interviewing is one of the most common and crucial ways that analysts obtain their information about a systems project. Much time is spent interviewing people about past projects they have worked on, how they have used those projects and any processing that might help add to their current and future work. During the interviewing process it is up to the interviewer to gather any information they feel would be a good fit for the individual working on their system. These features include anywhere from emotions, body language, facts and opinions.

You will want to first set up an appointment for the interviewee as well as provide a few questions and topics to think about in preparation for the interview. You should also prepare a guide for yourself on how much time you want to spend during the interview as well as each section of questions you have prepared for the interviewee. During the interview you also want to write down notes for yourself about the interviewee such as job history and any experience they may have that relates to the position and project at hand.

When it comes to asking the interviewee questions, you want to avoid asking “yes and no” questions and stick more to open-ended and close-ended questions as they are the best way to obtain information without having to ask a specific question. Some examples of open-ended questions this are as follows:

* + How do you prioritize tasks when you have multiple deadlines and dependencies?
  + Give an example of a time you had to work closely with a team to deliver a project. What was your role and how did you contribute to the team’s success?
  + How do you handle failure in a project?

Some examples of close-ended questions are as follows:

* + Are you comfortable with writing SQL queries and working with databases?
  + Have you worked with any cloud platforms such as Azure, GCP, or AWS?

During the interview you want to take notes about what the interviewee is saying as some of the information obtained could be vital to the project. If you have a chance and the interviewee consents, recording also helps. After the interview, you will want to go over your notes, interview multiple candidates, compare all their skills and then make your decision after reviewing.

Interviewing multiple candidates at once is also a possibility where several key candidates are interviewed at once. There are a few advantages to this such as it reduces the amount of time needed to interview people throughout the entirety of the interview process as well as creates synergy for groups being able to hear each other’s ideas. The disadvantages to this, however, are that the process is not entirely an efficient one and new interviews require coming up with new questions to ask.

* Observations: The observation step in the gathering process is exactly as it sounds. You will be directly observing workers at their jobs to see how they perform their day-to-day tasks, how they interact with information systems as well as other employees. This process is more accurate than an interview because the individual cannot hide or make up false information about what they do at their job since you are watching them perform in real time.

However, with this comes the fact that they may change their behavior on how they interact by knowing that they are being observed. The biggest downside to this process is that you are only looking from the outside in, on their behavior as it is not a full capture of what the individual does daily. It is human nature for people to change how they act or complete a process when they know they are being watched. Some may work faster than they usually do to show they perform more work in a day, and some may just become nervous and not perform as well.

An example of this process would be an employee stating how much time they spend running SQL queries in a database and how most of their day is spent on one specific task that may need to run queries. However, if you can view what the employee is doing in real time whether it be virtually or sitting with them, you may come to find that maybe they only actually run a few small queries daily that take up no more than 15 to 20 minutes to complete.

* Documentation: The documentation step in the gathering process is used to discover more details on systems that an organization supports. This process includes looking into any written documents on the systems they utilize as well as systems that may be under redesign. These documents can include anything from business plans, organization charts, business policy manuals, and even organizational mission statements. The analysis of these documents could tell you a lot about the requirements of a new system such as the following:
  + Problems with existing systems such as missing information or unneeded steps.
  + The reason why some systems are developed is the way that they are.
  + Opportunities to meet a system needs with key information
  + Systems that require certain information that might not be readily available

This information is useful in the sense that they can be features of current software that can lead to improvements for future software as well as identifying processing rules that must be enforced for any given reason.

There are two systems used when analyzing document:

1. Formal system: Which is the official way a system works as described in the organizations documentation.
2. Informal system: The way a system works.

There are also four major documents that are considered when creating a new system. Those steps are as follows:

1. Written work procedure
2. An invoice, which gives important information about how the organization operates.
3. A comparison of observation, which is used to determine what data is useful to capture
4. Documents that described how a system works as well as how it is used, such as user manuals.

# Context Diagram

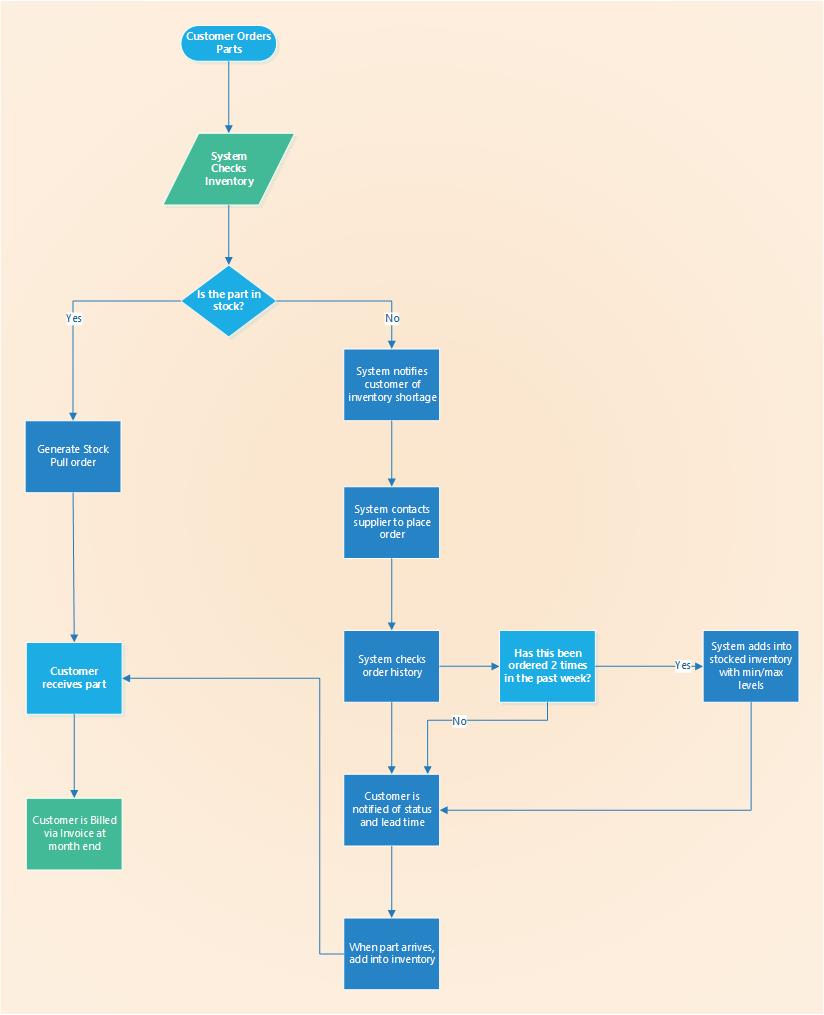
[](ISM3232/Denzel%20Coleman%20UML%20Diagram.vsdx)

Figure 3 - Context Diagram

# Use Case

A diagram of a person

Description automatically generated

Figure 4 - Receiving Use Case Diagram

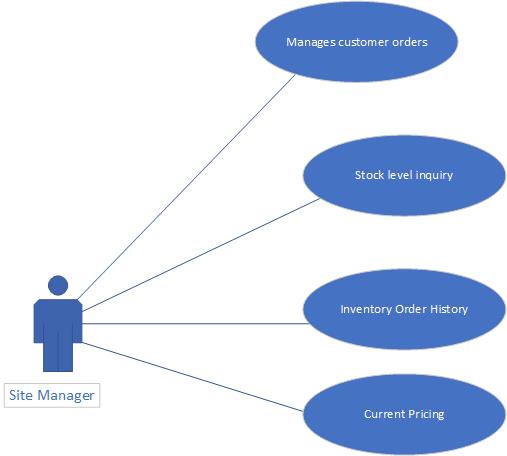


Figure 5 - Site Manager Use Case Diagram

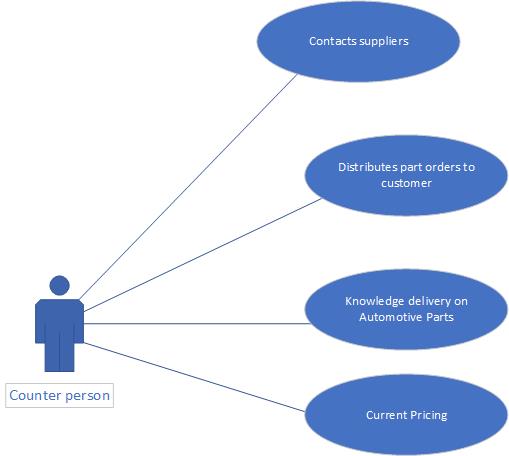


Figure 6 - Counter Person Use Case Diagram

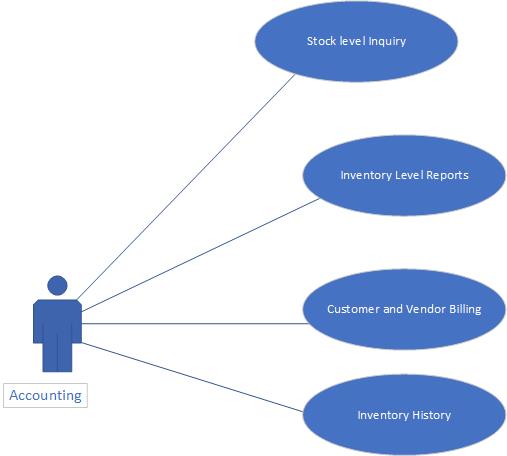


Figure 7 - Accounting Use Case Diagram

# Dialogue Diagram

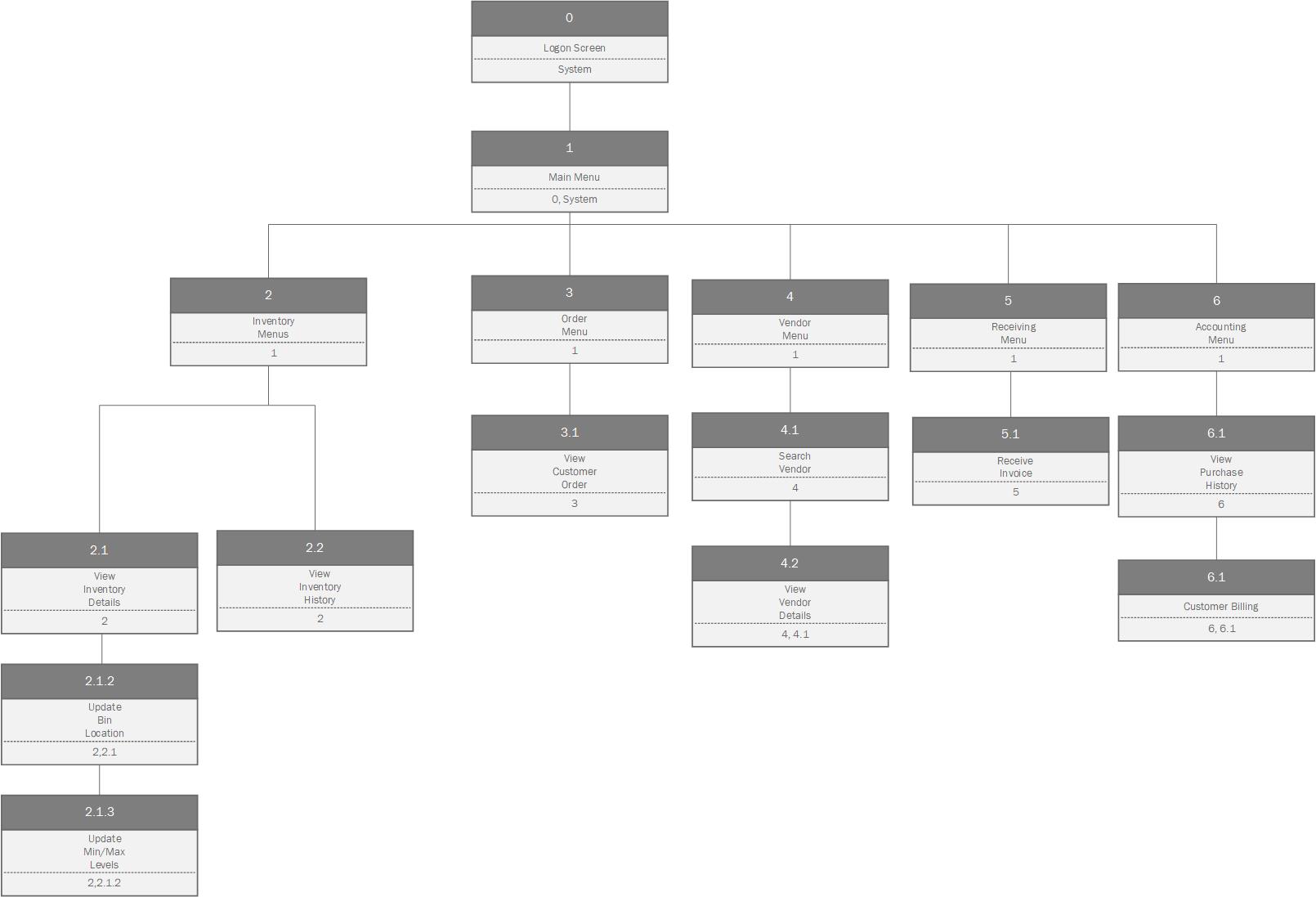


Figure 8 - Dialogue Diagram

# Prototypes



Figure 9 - Prototype Logon Page



Figure 10 - Prototype Inventory Page



Figure 11 - Prototype Order Page



Figure 12 - Prototype Vendor Page

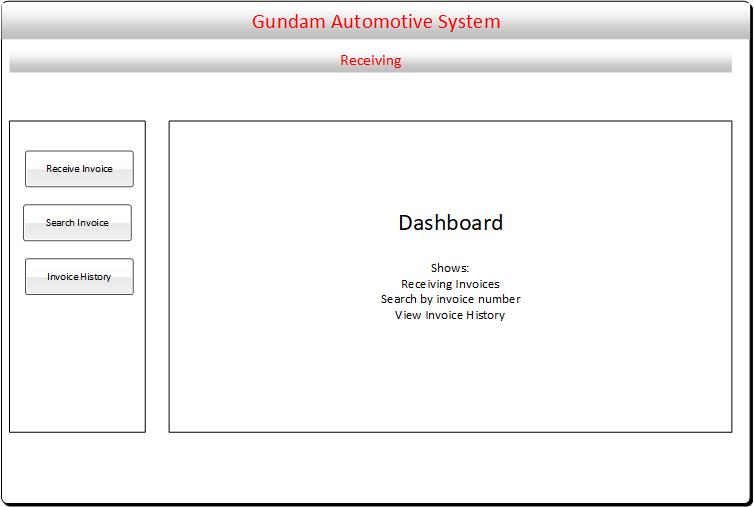


Figure 13 - Prototype Receiving Page

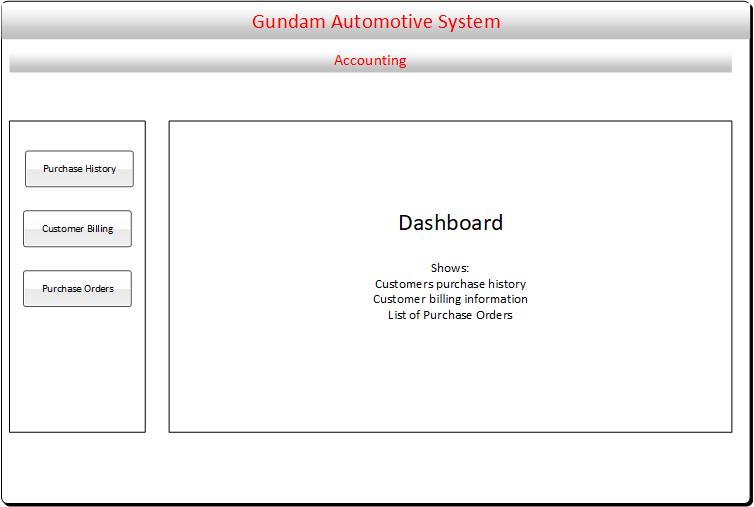


Figure 14 - Prototype Accounting Page

# Implementation Plan

* The purpose of the implementation process is to convert physical system specifications into software that can run it reliably, document any work that has been done and provide help for current and future users. The six main activities that make up system implementation and how they will be utilized in this software is as follows:

1. Coding: The process where the physical design specifications are converted into computer code by the team responsible for coding this software.
   * For this software, the team in charge of coding will be responsible for generating documentation for each individual module so that any programmer that is hired for maintenance can understand clearly how the program should work.
2. Testing: This process can work parallel with the coding process. With each module or major line of code that is created, the program can be tested to ensure it is working properly. Testing while the program is being coded early on is crucial to ensure there are no issues towards the end of the project.
   * For this software, the programmers will be testing each line of code as it is being written to ensure there are no errors and that all data is being pulled accurately. For documentation, the team will be listing the types of tests conducted, how it was used and how the system responded to the test no matter the result of passing or failing.
3. Installation: This is the process in which the current system is replaced by the new system. This includes the conversion of existing data, software, and work procedures that are in relation to the new system.
   * In this software, technicians will install the new system software as well as hardware at our location as well as provide documentation for setup and maintenance for both. New monitors and computer desktops are included in the project budget to help the flow of the process. It will be a hard habit to form for users that utilize other methods of getting the job done the older way but, in the end, it will be more effective for everyone.
4. Documentation: This is the crucial process that includes preparing documents and revealing all important information in relation to the creation and usage of the system software. The two audiences for this final documentation are the personnel that will be maintaining the system throughout its lifetime as well as the people utilizing it daily.
   * The way we will utilize the documentation process for this software will be to provide all documentation that was used to create this software to our users since they will be the ones using it daily. The information will be readily available to future and current users as they need so they can successfully navigate with no issues. Our information systems team will also have this information readily available so that if our users have any issues, they can guide and troubleshoot them virtually.
5. Training: In this process, analysts will be responsible for working with corporate the provide training tailored to the software that was developed. Users will be trained by corporate to understand how to use the system in the form of documentation.
   * For our company, we will be providing documentation both physically and digitally for each of our users to show step-by-step guidance on how to utilize this software. Since we are a smaller organization, our frontline associates will be the best way for we will all learn to use this software through trial and error.
6. Support: During this process, companies will provide ongoing educational and problem-solving assistance to information system users. Support materials and jobs will be designed as part of the implementation process. A help desk throughout the company is a single point of contact for any inquiry that is submitted about this software for all users.
   * Again, since our company is a smaller organization, any technical issues that our users come across that need fixing will have to be addressed to our IT team that can investigate these issues remotely during working hours to fix. Any issue that does come up will be important for our team to fix as soon as possible so that we can provide the best service to our clients.

# Maintenance Plan

* The Maintenance phase is the final phase within the software development lifecycle. By maintenance this means anything that requires the fixing or enhancing of an information system and consists of four main activities that occur within:

1. Corrective maintenance: Refers to changes made to repair any issues or defects in the design, coding or implementation of the system.
   * Corrective maintenance unfortunately repairs issues that come from design and programming errors. Because of this, we will need to ensure thorough testing within our plan and stress test the system through daily user interaction. Our associates will provide us with issues they have encountered along the way, report it to our information systems team, who will then have to get in contact with the analyst who created the software, enabling them to correct any of those issues.

1. Adaptive maintenance: Involves making changes to an information system to improve upon its functionality to changing business needs or the need to migrate it to another operating environment.
   * Within adaptive maintenance, our company will receive feedback from our information systems team to see how the current hardware will test against the constant usage of the new software. If any updates need to be made to help the system run better, we will implement the necessary changes to do so. We will also test the need for laptops for the associates and ensure that they can be set up at home in the event of an emergency for our clients as well as in the event of a hurricane since our location calls for bad weather.
2. Perfective maintenance: Involves making enhancements to improve processing performance or interface usability. It also involves the adding of system features that are not so much a necessity but more so a desire.
   * Since our associates will be the primary users of this system, we will be receiving their feedback on how the system performs and what issues they are encountering. We will also be asking for their feedback on what they think might help the system run better, what it can do without or what quality of life changes can be made to the existing code that works.
3. Preventive maintenance: Involves changes made to a system to reduce the chance of future system failure.
   * For this type of maintenance, we will be taking any feedback from our associates that resulted in failure or errors and work upon it within our IT team so that they can be avoided in the future. This includes any errors that are encountered with the system such as increasing the number of records that can be processed or even any difficulties with the hardware that need to be addressed such as a defective hard drive or internet connection issues.