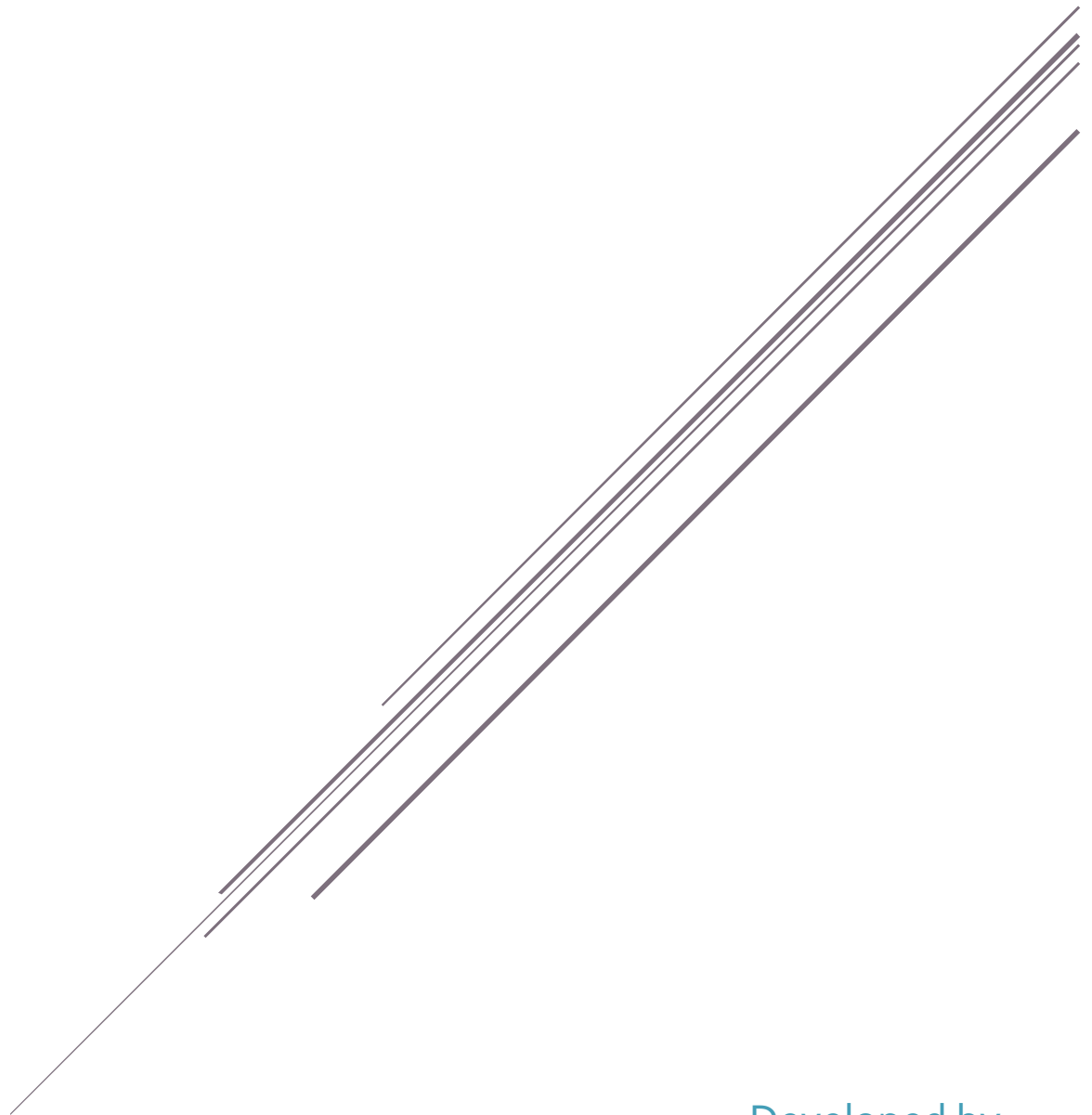


# COMPCARE

PC Repair Management System



Developed by:  
Ryan DeLosh, Salem Shaheen, Christopher Nuccio

## TABLE OF CONTENTS

<b><i>Abstract</i></b> _____	<b>3</b>
<b><i>Motivation</i></b> _____	<b>3</b>
<b><i>Significance</i></b> _____	<b>3</b>
<b><i>Related Work/Background</i></b> _____	<b>4</b>
<b><i>Proposed Work and Methods</i></b> _____	<b>5</b>
<b>OBJECTIVES:</b> _____	<b>5</b>
Problems: _____	<b>5</b>
Objectives: _____	<b>5</b>
<b>METHODOLOGY:</b> _____	<b>6</b>
<b><i>What will we do to solve these problems?</i></b> _____	<b>7</b>
<b>ALTERNATES/FALLBACK STRATEGIES:</b> _____	<b>8</b>
<b><i>Resources</i></b> _____	<b>8</b>
<b><i>Timeline/Work Plan</i></b> _____	<b>9</b>

## ABSTRACT

CompCare is a ticket management system specially designed for independent repair shops within the computer repair industry. CompCare helps its users keep track of system repairs and allows for a more consistent tracking system thus making its users more time efficient which then maximizes their profits.

## MOTIVATION

Many independent computer repair shops struggle with keeping up with the demand of the computer repair market. As computers get more sophisticated individuals don't know how to fix them and troubleshoot problems themselves so they bring them to a repair shop for diagnosis and repairs. Computer repair shops want to maximize their profits and pull ahead of their competition, to do so they need to get through system repairs fast and efficiently but many don't want to spend a lot for a commercial grade solution. We plan to provide an affordable solution, that is catered to the pc repair industry. It will be a barebones system that will include what required components to help organize the technicians time and repair schedule, thus maximizing their profits and improving their customer base. CompCare will have a simple straight forward user interface which allows for quick execution and management of computer repairs. Since CompCare will be simple and easy to use it will also can also possess the opportunity of being adaptable to other industries. If other non-computer repair businesses, see CompCare and want to adapt it to their industry they may submit a request for a modified version of CompCare to meet their needs.

## SIGNIFICANCE

CompCare is a straight forward system, it only has what is needed and nothing more thus reducing clutter and maximizing the technicians time and keeping a low learning curve for easy deployment into the industry. This will save technicians time and money which will then benefit their

customers and create a strong relationship between our customers (The repair shops) and their customers. CompCare aims to create a simple system that will help maximize our customers time and efficiency.

## RELATED WORK/BACKGROUND

Some related works are Zendesk, osTicket and Samanage. Zendesk is a ticket management system build for businesses and is designed around providing a customer service solution which in turn builds better customer relationships. Zendesk does not provide a streamlined system for in-house repair management it is more designed around allowing their users to provide their customers with an easy to adopt support system.

osTicket is a free open source ticket management system designed around customer support management just like Zendesk. Even though it is a free solution it osTicket does not provide a comprehensive suite designed for in house repair management, specifically on computer systems. CompCare will originally be a free system offered to its customers to use within their stores. Later it will be expanded and additional features can be added at a cost but not necessary for its basic features to work.

Samanage provides its customers with a program that is centered around remote IT assistance and allows for remote connections to the users customers that way the user can remotely fix their customers system via software. This does not provide a hardware repair solution unlike what CompCare aims to provide. CompCare is designed to provide a hardware repair management solution for technicians, it will help them manage their time and keep track of everything they are working with in a timely manner. Thus, making them more money because they are moving through repairs more efficiently.

## PROPOSED WORK AND METHODS

After having used ticketing software (such as Jira) and having built computer systems, we have a good idea of the design and flow needed to implement the CompCare project. We will pull ideas and features of these various ticketing software and apply them for the CompCare project. This means a ticketing system with a goal of product and customer order details. This is a straightforward system that is tailored specifically to the computer repair industry.

### OBJECTIVES:

We plan to create a web platform designed for computer service shops. This program will provide them with tools for scheduling customers system repairs and managing orders/jobs. This will require an interactive web interface, a database, a web server, and a system of communication between the database and server that is hosting the website. There are many issues we are looking to fix with our program.

### PROBLEMS:

- The first problem is that many small repair shops don't have the money to afford expensive systems for managing their repairs.
- The second problem is that many repair shops are not organized or systematic with their repairs. If their customer base increases their productivity decreases.
- The third problem is to keep a list of detailed notes electronically and easily searchable on what was done to complete the customer's order.

### OBJECTIVES:

- Create a responsive web interface that provides a straightforward queue system for ongoing system repairs.

- Create a web interface that's simple and easy to use with minimal learning curve.
- Create a website to manage computer system repairs and increase efficiency.
- Offer a minimal cost program that is straight forwards and contains what is needed for the target customer base
- System easily organizes orders and places them in order of when they need to get done by order of what's been waiting the longest.
- In each order in the system there is a detailed notes section that can be edited and viewed in case it needs to be referenced again in the future.

#### METHODOLOGY:

Our program will provide small business computer service shops with a simple service to keep track of customers and orders. We will be creating a program that allows users to:

- Add and edit computer orders with information / details like the following: Customer information, job (such as building a system, or fixing a hardware problem), and computer hardware / software specifications.
- CompCare will let employees be able to track each order with a status and details of each step of the order process (such as "Status: Building... computer mostly built, waiting for sata cables to come in").
- Sort through tickets by various categories, easily and cleanly (such as open / closed, by type of order, and priority).

## WHAT WILL WE DO TO SOLVE THESE PROBLEMS?

Our program will need to include the following parts:

1. A GUI Web Interface
2. A database for managing accounts
3. A database for user data
4. A database for customer data
5. Backend for communication / data retrieval
6. Provide in detail fields within a ticket for technicians to keep track of computer systems

(Sample list of fields that will be included in a ticket)

1. Customer Name
2. Computer description
3. Unique repair ID number (auto generated) {Number will be put on physical computer for reference}
4. Computer issue reported by customer
5. Diagnosis
6. Notes on repairs
7. Technician working on computer
8. Expense list
9. Date in and date out of shop (if complete)

## 10. How problem was solved

We plan to make use of the following development technologies:

1. HTML
2. CSS
3. JavaScript
4. Node.js
5. MySQL or Mongo (Not sure which one yet)

### ALTERNATES/FALLBACK STRATEGIES:

We plan to use MongoDB as our database system but if we can't get that to work we will revert to using MySQL. If we use MySQL as the database, we will use AWS for database hosting instead of mLab since mLab is only a MongoDB hosting service. There is also the option of using Google's Firebase software suite which also includes database hosting. We will be sticking with standard JavaScript and jQuery with this assignment, so we can get familiar with JavaScript and how it works before moving to a framework.

### RESOURCES

mLab: <https://mlab.com/> (MongoDB hosting)

Heroku: <https://www.heroku.com/> (Backend Hosting)

Either AWS or WebLab: (Front End Hosting)

jQuery: <https://jquery.com/>



Bootstrap 4: <https://getbootstrap.com/>

## TIMELINE/WORK PLAN

See Attached Excel Spreadsheet / pdf