Nottingham Trent University

School of Science and Technology

Customer Relationship Management System

by

Alvin Sung

in

2018

Project report in part fulfilment

of the requirements for the degree of

Bachelor of Science with Honours

in

Computing

I hereby declare that I am the sole author of this report. I authorize the Nottingham Trent University to lend this report to other institutions or individuals for the purpose of scholarly research.

I also authorize the Nottingham Trent University to reproduce this report by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Alvin Sung

# Abstract

Customer Relationship Management (CRM) is a system that businesses use which revolves around customers. It includes many features that aid companies in gaining a long run of profitability and more rewards. In order to enhance these features in the CRM system, this project is about developing a Customer Relationship Management (CRM) system in C#. With businesses growing and most of businesses sell similar products to others and businesses will need a management of their core business processes. The Enterprise Resource Planning (ERP) system is created to manage the core business processes and CRM is a one of the core systems in an ERP system. The main purpose of CRM is to improve, maintain the relationship with customers or attract potential customers by using data analysis, automation, customer service and many other features.

The objective of this project is developing a CRM system with solutions to limitations and improve on some features already presented on current CRM systems. Since that CRM software are constantly being improved and businesses will have to keep changing CRM systems to gain certain benefits that others do not which costs organisations a lot of time and money in transition to different CRM systems.

CRM in this project include the main modules that all CRM systems have, which

The CRM that has been created includes the main features that all CRM systems have, which include Contact management, dashboards and reports, email implementation, Social media management, inventory management, automation and much more. Since CRM is focused on customers, there will be customer filtering and customer profiling with promotion tools. The tools used to develop this CRM include C# and Visual Studio, OleDB plug in and Microsoft Access.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr Tawfik Al-Hadhrami, throughout the duration of completing this final year project. This project would not have succeeded without his guidance and assistance. The project has helped me gain valuable knowledge about Customer Relationship Management (CRM) system.

I would like to extend my thanks to my family and friends for their support throughout the year and encouraging me to do as well as I can throughout university.

Finally, I would like to thank all the lecturers who had taught me in previous years. Thanks to them, I had learned valuable skills which I could incorporate into this final year project.

# Table of Contents

Table of Contents

[Abstract 3](#_Toc512407582)

[Acknowledgements 4](#_Toc512407583)

[Table of Contents 5](#_Toc512407584)

[List of Tables 8](#_Toc512407585)

[List of Figures 9](#_Toc512407586)

[List of Abbreviations 11](#_Toc512407587)

[1. Introduction 12](#_Toc512407588)

[1.1. Motivation and Problem Statement 12](#_Toc512407589)

[1.2. Aim of Project 12](#_Toc512407590)

[1.3. Project Overview 12](#_Toc512407591)

[1.4. Project Objectives 13](#_Toc512407592)

[1.5. Project Management 13](#_Toc512407593)

[2. Literature Review 15](#_Toc512407594)

[2.1. Introduction 15](#_Toc512407595)

[2.2. Customer Relationship Management 15](#_Toc512407596)

[2.2.1. Definition of CRM 15](#_Toc512407597)

[2.2.2. Types of CRM 16](#_Toc512407598)

[2.2.3. Relationships 20](#_Toc512407599)

[2.2.4. Advantages of CRM 21](#_Toc512407600)

[2.2.5. Limitations to CRM 22](#_Toc512407601)

[2.2.6. Current Solutions by other Companies 22](#_Toc512407602)

[2.2.7. List of Features by other companies 25](#_Toc512407603)

[3. New Ideas 32](#_Toc512407604)

[3.1. Proposed System 32](#_Toc512407605)

[3.2. Data Flow Diagram 33](#_Toc512407606)

[3.3. Tools 34](#_Toc512407607)

[3.4. Features compared to other companies 36](#_Toc512407608)

[3.5. Proposed Layout Design 40](#_Toc512407609)

[3.5.1. Dashboard 40](#_Toc512407610)

[3.5.2. Contacts Form 41](#_Toc512407611)

[3.5.3. Calendar Form 42](#_Toc512407612)

[3.5.4. Tasks Form 43](#_Toc512407613)

[3.5.5. Inventory Management 44](#_Toc512407614)

[3.5.6. Email Form 45](#_Toc512407615)

[3.5.7. Customer Form 46](#_Toc512407616)

[3.6. Data Dictionary 46](#_Toc512407617)

[4. Implementation 49](#_Toc512407618)

[4.1. Interface Design 50](#_Toc512407619)

[4.2. Source Code 56](#_Toc512407620)

[4.2.1. Chart Control 56](#_Toc512407621)

[4.2.2. Google Drive 57](#_Toc512407622)

[4.2.3. Insert into Access Database 63](#_Toc512407623)

[4.2.4. Read Data from Access Database 64](#_Toc512407624)

[4.2.5. Update Data to Access Database 65](#_Toc512407625)

[4.2.6. Delete Data from Access Database 66](#_Toc512407626)

[4.2.7. Search Data grid view feature 66](#_Toc512407627)

[4.2.8. Form Changes 67](#_Toc512407628)

[5. Results and Discussion 68](#_Toc512407629)

[5.1. System Testing 69](#_Toc512407630)

[5.1.1. Dashboard 69](#_Toc512407631)

[5.1.2. Contacts Form 73](#_Toc512407632)

[5.1.3. Calendar Form 84](#_Toc512407633)

[5.1.4. Tasks Form 90](#_Toc512407634)

[5.1.5. Inventory Management Form 99](#_Toc512407635)

[5.1.6. Email Form 107](#_Toc512407636)

[5.1.7. Customers Form 110](#_Toc512407637)

[5.2. Implementation Issues and Challenges 112](#_Toc512407638)

[6. Conclusion/Future works 113](#_Toc512407639)

[6.1. Successful Outcomes 113](#_Toc512407640)

[6.2. Unsuccessful Outcomes 113](#_Toc512407641)

[6.3. Future Enhancements 113](#_Toc512407642)

[References 115](#_Toc512407643)

[Bibliography 117](#_Toc512407644)

# List of Tables

[Table 1 - Camparison of 3 most popular CRM systems 31](#_Toc512407566)

[Table 2 - My CRM compared to Others 39](#_Toc512407567)

[Table 3 - Contacts Database Data 47](#_Toc512407568)

[Table 4 - Events Database Data 48](#_Toc512407569)

[Table 5 - Tasks Database Data 48](#_Toc512407570)

[Table 6 - Inventory Database Data 48](#_Toc512407571)

[Table 7 – Item Sales Database Data 49](#_Toc512407572)

[Table 8 - Monthly Sales Database Data 49](#_Toc512407573)

[Table 9 - Customer Database Data 49](#_Toc512407574)

[Table 10 - Test table headings 69](#_Toc512407575)

[Table 11 - Dashboard Form Test Table 70](#_Toc512407576)

[Table 12 - Contacts Form Test Table 75](#_Toc512407577)

[Table 13 - Calendar Form Test Table 86](#_Toc512407578)

[Table 14 - Tasks Form Test Table 92](#_Toc512407579)

[Table 15 - Inventory Management Form Test Table 101](#_Toc512407580)

[Table 16 - Email Form Test Table 108](#_Toc512407581)

# List of Figures

[Figure 1: Flowchart of Methodology for Final Year Project 14](#_Toc512407510)

[Figure 2 - CRM Elements 16](#_Toc512407511)

[Figure 3 - Automations that are included in operational CRM 17](#_Toc512407512)

[Figure 4 - Sales increasing by acquiring new customers and dealing with existing customers 18](#_Toc512407513)

[Figure 5 - Elements of marketing modules included in marketing automation 18](#_Toc512407514)

[Figure 6 - Service modules included with service automation 19](#_Toc512407515)

[Figure 7 - Salesforce Logo 23](#_Toc512407516)

[Figure 8 - Zoho Logo 23](#_Toc512407517)

[Figure 9 - Apptivo Logo 24](#_Toc512407518)

[Figure 10 - System Flow Diagram 34](#_Toc512407519)

[Figure 11 - Proposed Dashboard Layout 40](#_Toc512407520)

[Figure 12 - Proposed Contacts Form Layout 41](#_Toc512407521)

[Figure 13 – Proposed calendar form layout 42](#_Toc512407522)

[Figure 14 - Proposed tasks form layout 43](#_Toc512407523)

[Figure 15 - Proposed Inventory Management Form 44](#_Toc512407524)

[Figure 16 - Proposed email form layout 45](#_Toc512407525)

[Figure 17 - Proposed Customer Form Layout 46](#_Toc512407526)

[Figure 18 - Header Design 50](#_Toc512407527)

[Figure 19 - Navigation Design 51](#_Toc512407528)

[Figure 20 - Dashboard Form 52](#_Toc512407529)

[Figure 21 - Contacts Form 53](#_Toc512407530)

[Figure 22 - Calendar Form 53](#_Toc512407531)

[Figure 23 - Tasks Form 54](#_Toc512407532)

[Figure 24 - Inventory Form 55](#_Toc512407533)

[Figure 25 - Email Form 56](#_Toc512407534)

[Figure 26 - Chart change test with dummy data 71](#_Toc512407535)

[Figure 27 - Chart change test with dummy data 72](#_Toc512407536)

[Figure 28 - Add Function Test 75](#_Toc512407537)

[Figure 29 - Save Button pressed when textboxes are filled with data 76](#_Toc512407538)

[Figure 30 - Data inserted into database and loaded into data grid view 77](#_Toc512407539)

[Figure 31 - Selected record data outputted to correct textboxes 78](#_Toc512407540)

[Figure 32 - Edit function test 79](#_Toc512407541)

[Figure 33 - Data being changed in textbox 80](#_Toc512407542)

[Figure 34 - Update function test 81](#_Toc512407543)

[Figure 35 - Delete Function test. 82](#_Toc512407544)

[Figure 36 - Search Function test 83](#_Toc512407545)

[Figure 37 - Add button pressed test 86](#_Toc512407546)

[Figure 38 - Data inputted in textboxes and date and dropdown list is selected 87](#_Toc512407547)

[Figure 39 - Save button press Test 88](#_Toc512407548)

[Figure 40 - Delete button pressed test 89](#_Toc512407549)

[Figure 41 - Add Button Pressed Test 92](#_Toc512407550)

[Figure 42 - Data inputted into textbox and Dates selected 93](#_Toc512407551)

[Figure 43 - Save Button Pressed Test 94](#_Toc512407552)

[Figure 44 - Edit Button Pressed Test 95](#_Toc512407553)

[Figure 45 - Update Button Pressed Test 96](#_Toc512407554)

[Figure 46 - Delete Button Pressed Test 97](#_Toc512407555)

[Figure 47 - Record Deleted in data grid view 98](#_Toc512407556)

[Figure 48 - Add Button Pressed Test 101](#_Toc512407557)

[Figure 49 - Data inputted into textbox and Dates selected 102](#_Toc512407558)

[Figure 50 - Save Button Pressed Test 103](#_Toc512407559)

[Figure 51 - Edit Button Pressed Test 104](#_Toc512407560)

[Figure 52 - Update Button Pressed Test 105](#_Toc512407561)

[Figure 53 - Delete Button Pressed Test 106](#_Toc512407562)

[Figure 54 - Email inbox in Data grid view 108](#_Toc512407563)

[Figure 55 - Email Content Reader Test 109](#_Toc512407564)

[Figure 56 - Filter Data through Search 111](#_Toc512407565)

# List of Abbreviations

**CRM** - Customer Relationship Management

**GUI** - Graphical User Interface

**DBMS** - Database Management System

**.snl** - Microsoft Visual Studio Solution

**ASE** - Advanced Semiconductor Engineering Inc.

**SFA** - Sales Force Automation

**MA** - Marketing Automation

**SA** - Service Automation

**B2C** - Business to Consumer

**B2B** - Business to Business

**AI** - Artificial Intelligence

# Introduction

## Motivation and Problem Statement

CRM systems give a lot of benefits to companies, but they also provide many challenges. These challenges stop companies from fully using CRM systems to its utmost potential. “Tools and workflows can be complex to implement, especially for large enterprises. Previously these tools were generally limited to contact management: monitoring and recording interactions and communications.” () A lot of organisations have failed to meet the customer’s expectations due to multiple of reasons such as misallocation of resources, not setting achievable objectives and not as much support from the company. CRM are continuously evolving and are certainly not perfect. CRM will need to evolve with the current trends with the latest trend being 2-3 years ago latest, but they still do not find solutions to some of the limitations that have been found as Dimitrova (2017) mentions “Customer Relationship Management software (CRM) hasn’t changed a lot since the 1990s, except that it moved from on-premise systems to the cloud.”. By failing to meet customer’s set expectations will mean that the company will lose customers and reputation and losing customers will mean losing revenue. Therefore, CRM need to be modern and dynamic by using other features such as automation, tracking customer statistics etc. With the correct and working CRM can help the company to identified, acquire, satisfy and retain profitable customers from existing customer base.

## Aim of Project

This project aims to create a Contact Management CRM software for every size and type of business which would benefit them with functions that other CRM software can do, such as retrieving and storing data but with more features like being able to purchase additional features of the software to make sure that businesses do not have to switch CRM software to benefit the new requirements and have to only stick with one software to reduce cost and stress to employees, whilst also implementing solutions to the limitations of CRM.

## Project Overview

In this project, research on customer relationship management (CRM) will be conducted to determine the differences between the CRM’s that are currently out and their functionalities they provide. After will be creating the prototype basing on the research that has been found and try to create solutions to the limitations that have been found during the research phase. Once the prototype has been completed, it will be tested and modified until submission deadline.

## Project Objectives

This project will have a product that will be able to do what every CRM is able to do and allow simple analysis of data. For example, the user will be able to view graphs of sales ranging from current to the past 6 months and inventory.

The CRM software shall be able to create, read, update and delete data from a database which will be Microsoft access. The data being stored and retrieved shall be contacts, employees and customer details to increase efficiency so that several users will be able to share information to different departments.

The CRM software shall allow information to be shared across departments on customer problems, buying habits etc.

The CRM shall provide a promotion management. The promotion management should include create, modify and deliver the promotion to customers.

The CRM software shall provide additional feature integrated in, such as email and dialling capabilities. There will be other options for the CRM for companies that will allow the user/buyer to decide whether they would like the extra integrated functions to be accessible rather than having everything available from the start.

## Project Management

The initial project outline is shown in Figure 1 below. A review point will be completed half way through the project time allowed to review how the work was progressing and what need to be done.

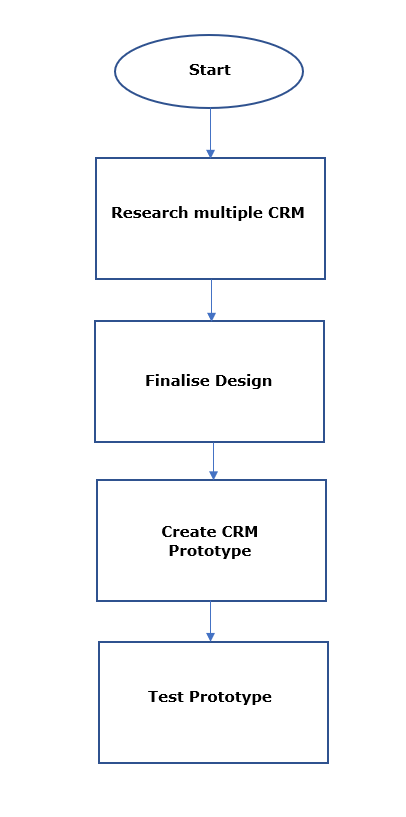


Figure : Flowchart of Methodology for Final Year Project

# Literature Review

## Introduction

This literature review for the project will be reviewing articles, books and internet resources to gather information about CRM, automation, limitations and current CRM software.

## Customer Relationship Management

## Definition of CRM

According to Buttle (2009), the definition of CRM is “CRM is the core strategy that integrates internal processes and functions, and external networks, to create and deliver value to targeted customers at a profit. It is grounded on high quality customer-related data and enabled by information technology”. Whereas, Brown (1999) state that “Customer Relationship Management (CRM) is neither a concept nor a project. Instead, it’s a business strategy that aims to understand, anticipate and manage the needs of an organization’s current and potential customers.” And Hu (2015) also define CRM as " Customer Relationship Management(CRM) is a management concept originated in relationship-marketing.”. This means that CRM is just a strategy to aim to get revenue and profit from current and potential customers, as according to Prayitno (2017) mentions “CRM is a customer-centric business strategy which a company employs to improve customer experience and satisfaction by customizing products and services to customers’ needs.”. To incorporate the strategy and make it easier and much better in present times, CRM systems have been created with the advancing technology.

A popular company who creates CRM systems, Salesforce mentions “A CRM system helps you keep your customer’s contact details up to date, track every interaction they have with your business, and manage their accounts. It’s designed to help you, improve your customer relationships, and in turn, customer lifetime value.” Therefore, a CRM is used to manage business relationships from customers to clients to employees that the company has and to help grow the business in a long term. “CRM is expanded as customer relationship management, which helps to analyse and mange customer history with an enterprise and use the knowledge to increase the business.” (Prabha and Subramanian,2017) As it is believed that it is easy to gain new customers but more difficult to maintain customers to the company, CRM is solely focusing on customers in the aspect of acquisition and retaining customers which in return would increase the sales growth. “treating existing customers well is the best source of profitable and sustainable revenue growth.” (Brown, Coopers 1999) CRM elements are shown in figure 2 (tutorialspoint, 2014) (Prabha and Subramanian,2017).

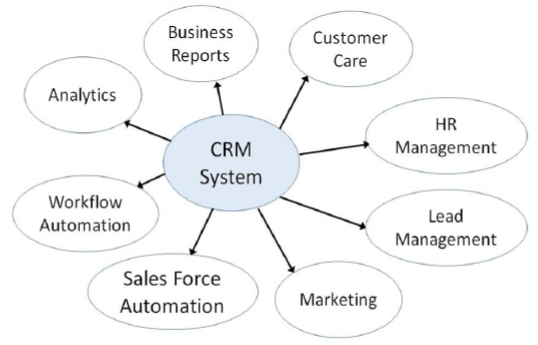


Figure - CRM Elements

## Types of CRM

There are four types of CRM systems that are used depending on what the organization are doing/need. The following list are the type of CRM systems with an explanation of what that type specialises in doing and their characteristic features:

* Operational – This type of CRM is “based on customer-oriented processes, such as selling, marketing, and customer service.” (tutorialspoint, 2014) and according to Prabha and Subramanian (2017) operational CRM “provides single view of customer information, which includes sales force, service and marketing automation.” As shown in figure 3 (tutorialspoint, 2014). Whereas Buttle (2009) states “Operational CRM automates and improves customer-facing and customer-supporting business processes.”

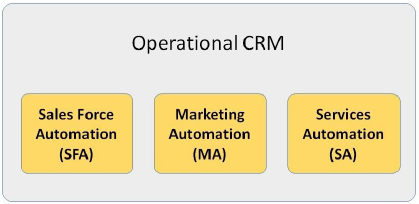


Figure - Automations that are included in operational CRM

* + Sales Force Automation – “Sales Force Automation (SFA) is the process of using software to automate sales in a business. SFA streamlines and automates process-heavy tasks such as inventory control, order processing, contact management, sales forecasting, analysis, and employee performance, freeing up resources to focus on increasing sales and revenue.” (SalesForce,). Therefore, SFA is manages the companies selling processes automatically such as, identification of needs, closing a sale etc. Whereas Buttle (2009) defines SFA as “Sales-force automation is the application of computerized technologies to support salespeople and sales management in the achievement of their work-related objectives.” Main purpose of sales automation is to set business standard within the organization to acquire new customers and deal with existing customers.” (TechOneStop) All of this is to increase total amount of sales to bring in more revenue and profit as shown in figure 4 (tutorialspoint, 2014)

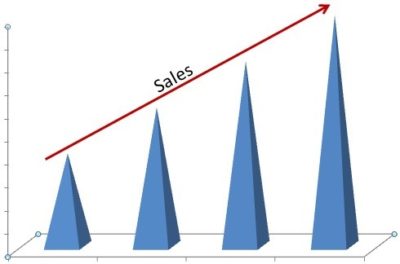


Figure - Sales increasing by acquiring new customers and dealing with existing customers

* + Marketing Automation – “helps the organization to streamline marketing process, automate and measure marketing task to increase sales and earn revenue faster. It includes various CRM marketing modules like campaign Management, Event Based Marketing.” (TechOneStop) and Buttle (2009) defines MA as “Marketing automation is the application of computerized technologies to support marketers and marketing management in the achievement of their work-related objectives.”. Elements of Campaign Management and Event Based marketing is shown in Figure 5 (tutorialspoint, 2014).



Figure - Elements of marketing modules included in marketing automation

* + Services Automation – “enables business to provide best quality of customer service by automating the service process. It includes various service modules like call management, case or incident management, knowledge management, service level management.” (TechOneStop) and Buttle (2009) mentions that “Service automation allows companies to manage their service operations, whether delivered through call centre, contact centre, web or face-to-face.” Service operations/processes are shown in figure 6 (tutorialspoint, 2014).



Figure - Service modules included with service automation

* Analytical – This CRM is “based on the intelligent mining of the customer data and using it tactically for future strategies” (tutorialspoint, 2014) This CRM focuses on customer-related data by capturing, interpreting, segregating, storing, modifying, processing, and reporting that data. According to Prabha and Subramanian (2017) they mention “with the analytical techniques like data mining techniques, analytical CRM applied to customer data to predict the future, present scope and to enhancing the better services to customers.”
* Collaborative – Collaborative CRM is “based on application of technology across organisation boundaries with a view to optimise the organisation and customers.” (tutorialspoint, 2014) According to Prabha and Subramanian (2017) collaborative CRM’s purpose is “to share the customer information across the department to make business process easier.”
* Strategic – A strategic CRM is “customer-centric, based on acquiring and maintaining profitable customers.” (tutorialspoint, 2014) This CRM is for businesses that puts customers first by collecting information about the customers and market trends, then segregating it to come up with a better value proposition. According to Buttle (2009) “Strategic CRM is focused upon the development of a customer-centric business culture.”. This means that it is more dedicated on winning over customers and keeping customers by creating and delivering values compared to other competitors.

## Relationships

Relationship is key between customer and a business. “A relationship is composed of a series of interactive episodes between dyadic parties over time.” (Buttle, 2009). This means that these interactive episodes are timed bound. A business will normally have a sales representative to sell a product to customers, which they would try to establish a relationship with customers or vice versa by the customer enquiring or purchasing a product etc.

Companies would want to build relationships with customers as it would benefit companies in gaining profit and building up the company as according to Buttle (2009) “The fundamental reason for companies wanting to build relationships with customers is economic.”. Buttle (2009) also states that “Companies generate better results when they manage their customer base in order to identify, acquire, satisfy and retain profitable customers. These are key objectives of many CRM strategies.” This means that companies improving the number of customers they retain will improve other factors such as, gaining new customers, more sales, reputation etc. with the main goal is being profit. Advantages of building a strong relationship and retaining customers include reduced marketing costs, as according to Buttle (2009) states “Improving customer retention reduces a company’s marketing costs.” Also, another advantage is better customer insight as Buttle (2009) states “As customer tenure lengthens, suppliers are able to develop a better understanding of customer requirements and expectations.”. This would mean that the relationship between companies and customers will deepen with trust and commitment will grow over time and will become loyal, which would mean that revenue and profit will be more steady and secure.

Companies may not want relationships with customers as it would bring disadvantages to them in a business to business context. According to Buttle (2009), there are 4 disadvantages which are as listed:

* Loss of control
* Exit costs
* Resource commitment
* Opportunity costs

Relationships aren’t only built between companies and customers, as relationship can also be built between customers and suppliers in a business to business context and a business-to-consumer context. Buttle (2009) lists 5 reasons for why a B2B customers might want a long-term relationship with a supplier:

* Product complexity
* Product strategic significance
* Service requirements
* Financial risk
* Reciprocity

Buttle (2009) lists 6 benefits for B2C relationships:

* Recognition
* Personalization
* Power
* Risk reduction
* Status
* Affiliation

## Advantages of CRM

CRM systems provide many benefits to organisations no matter the size of the organisation, type of products being sold or what industry the organisation based on, as the main focus of a CRM system are the customers. Nibusinessinfo (2017) mentions some advantages as listed:

* efficiencies, particularly in data management
* better collaboration and communication across departments and teams
* greater accountability
* improved customer experience
* better reporting and analysis capabilities

These advantages will help organisations in a financial aspect as they will be increasing sales to customers, identifying which customer is more profitable and selling alternative products than their main product.

In conclusion, CRM systems help organisations by

* Improve customer satisfaction
* Improve or maintain a good reputation in that industry
* Increase efficiency
* Total cost of sales reduced
* Business processes made easier
* Cost reduction in customer service.

## Limitations to CRM

A limitation in CRM that has been mentioned is automation. Automation being the future of CRM, there isn’t much automation in current CRM systems/software. Since Dimitrova (2017) mentioned that automation will be key in the future of CRM and that CRM has not changed much since the 1990’s except that the system is on the cloud rather than on-site premises.

The other limitations are stated by Hill (2005) are as listed:

* 1. Failure to adjust business process – Vendors speak about the technology but not the business processes; which then are not adjusted accordingly to implement the system at its optimal performance.
  2. Complexity – CRM projects are very complex which is the biggest hurdle. The implementation can consist of dozens of related goals and projects. The failure of some of these tasks may result in the entire project being called a failure when in fact it is more of an incomplete project than a failure.
  3. Customer Identification Parameters – Defining customers is another major hurdle that will need to be overcome to achieve a customer-focused organisation. For example, companies would segment their customers based on buying behaviours, demographics etc.

## Current Solutions by other Companies

Salesforce cloud by the company Salesforce is a popular and one of the top CRM’s created, which offers applications for all businesses sizes. Their CRM allows users to synchronise their contacts, calendars, emails and tasks from outlook. They have 2 editions which are for small business and large businesses. The small business edition has applications to manage contacts, track sales deals, manage tasks and events, harvest leads and track performance and the large business edition, has all the applications in the small business edition but with extra applications for call scripts, team-selling functionality, business workflow, setup approval and automation, custom applications, API integrations and more.

All the applications store and retrieve data from the cloud as Salesforce mention “Salesforce offers a wide variety of CRM categories and systems to meet your needs, including Sales Cloud, Marketing Cloud, Service Cloud, Analytics Cloud, Data Cloud, Community Cloud, App Cloud, and IoT, serving more than 150,000 customers.”

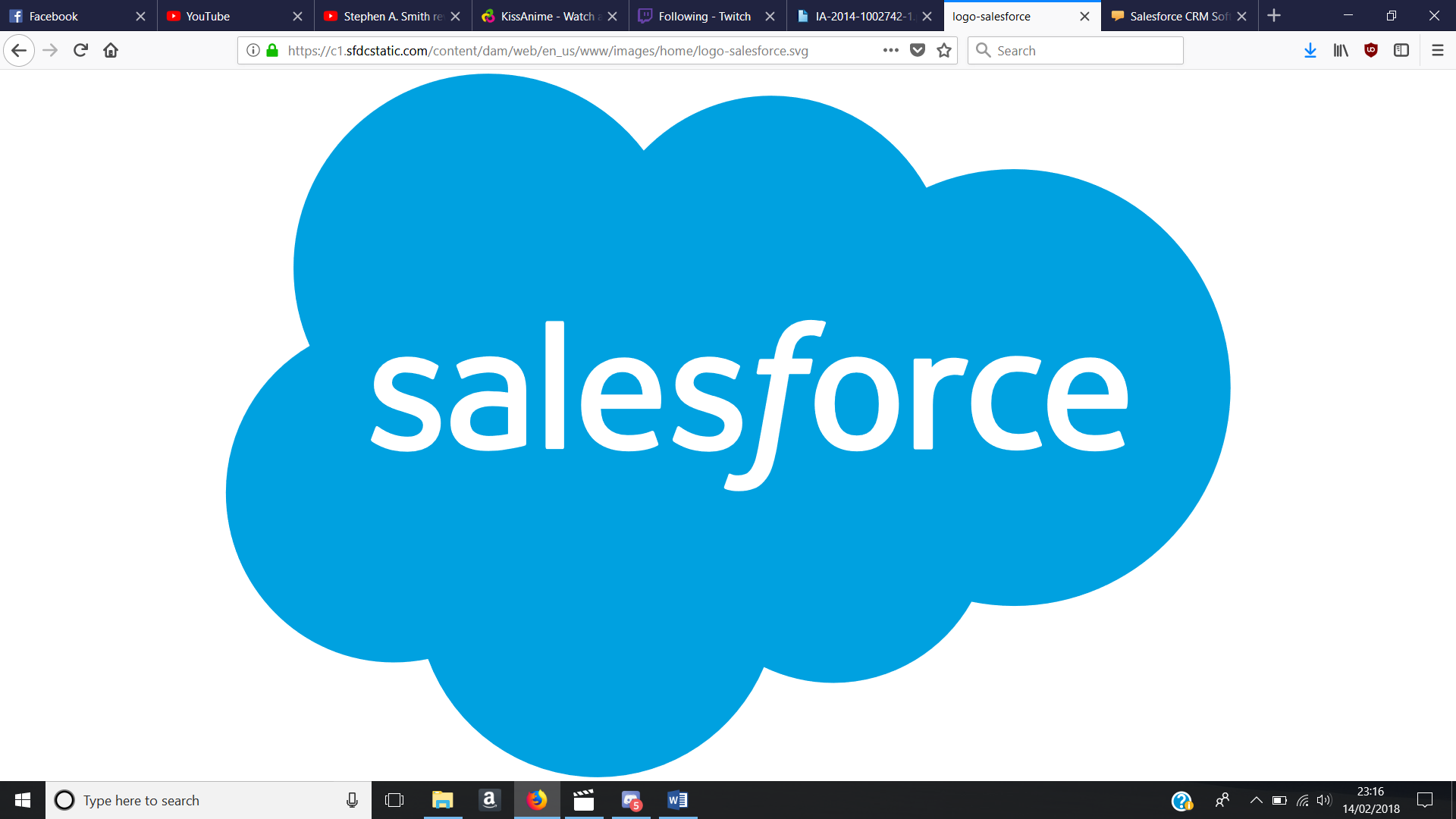


Figure - Salesforce Logo

Another popular CRM is Zoho CRM. It has multiple features like Salesforce ranging from Multichannel, Zia, Automation, Analytics, Customisation, Collaboration, Security and Developer tools. Zoho has included an AI feature which they call Zia. “Zia learns about your business and your team, it will offer predictions on trends, anomalies, conversions, and deals closing. Zia also *automatically automates* recurring tasks to speed up your sales cycle.” (Zoho 2018)



Figure - Zoho Logo

Apptivo CRM is another popular CRM. Apptivo has similar features as Zoho and Salesforce. It has G-suite integration which uses features such as google drive, google calendar etc. and very customizable for organisations.



Figure - Apptivo Logo

## List of Features by other companies

As there are a lot of different CRM systems that have been released by different companies, they will have features that are the same and others that have not been implemented. A comparison table between Zoho CRM, Apptivo CRM and Salesforce CRM is shown in table 1. GetApp (N.A) was used to compare features between Zoho, Apptivo and Salesforce CRM.

|  |  |  |  |
| --- | --- | --- | --- |
| **Product** | **Zoho CRM** | **Apptivo CRM** | **Salesforce CRM** |
| **Platform Supported** |  |  |  |
| Web based | ✓ | ✓ | ✓ |
| iPhone app | ✓ | ✓ | ✓ |
| Android app | ✓ | ✓ | ✓ |
| Windows Phone app | X | ✓ | X |
| **Typical Customers** |  |  |  |
| Freelancers | ✓ | ✓ | X |
| Small businesses | ✓ | ✓ | ✓ |
| Mid-size Business | ✓ | ✓ | ✓ |
| Enterprise | ✓ | ✓ | ✓ |
| **Support** |  |  |  |
| Phone support | ✓ | ✓ | ✓ |
| Online support | ✓ | ✓ | ✓ |
| Knowledge base | ✓ | ✓ | ✓ |
| Video tutorials | ✓ | ✓ | ✓ |
| **Features** |  |  |  |
| API | ✓ | ✓ | ✓ |
| Activity Dashboard | ✓ | ✓ | ✓ |
| Activity Tracking | ✓ | X | ✓ |
| Ad hoc Reporting | ✓ | X | ✓ |
| Approval Process Control | ✓ | X | ✓ |
| Auto-Responders | ✓ | ✓ | X |
| Automatic Reminders |  | ✓ | ✓ |
| Calendar Management | ✓ | ✓ | X |
| Calendar Sync with Google | ✓ | ✓ | X |
| Campaign Management | ✓ | X | ✓ |
| Configurable Workflow | ✓ | ✓ | ✓ |
| Contact Database | ✓ | X | ✓ |
| Contact Management | X | ✓ | ✓ |
| Custom Fields | ✓ | ✓ | X |
| Customer History | ✓ | ✓ | ✓ |
| Customizable Approvals | ✓ | X | ✓ |
| Customizable Reporting | ✓ | ✓ | X |
| Customizable Templates | ✓ | ✓ | X |
| Data Visualization | ✓ | X | ✓ |
| Document Storage | ✓ | ✓ | X |
| Drag & Drop Interface | ✓ | X | ✓ |
| Email Integration | ✓ | ✓ | X |
| Email Templates | ✓ | ✓ | ✓ |
| Email Tracking | ✓ | X | ✓ |
| Filtered Views | ✓ | ✓ | ✓ |
| History Tracking | ✓ | X | ✓ |
| Inbox Management | ✓ | X | ✓ |
| Lead Assignment | ✓ | ✓ | X |
| Lead Capture | ✓ | ✓ | X |
| Lead Distribution | ✓ | ✓ | ✓ |
| Lead Generation | ✓ | ✓ | X |
| Lead Management | ✓ | ✓ | ✓ |
| Lead Scoring | ✓ | ✓ | ✓ |
| Microsoft Outlook Integration | ✓ | X | ✓ |
| Mobile Integration | ✓ | X | ✓ |
| Multi-Currency | ✓ | ✓ | ✓ |
| Opportunity Management | ✓ | ✓ | ✓ |
| Pipeline Management | ✓ | ✓ | ✓ |
| Pipeline Reports | ✓ | ✓ | ✓ |
| Purchase Order Management | ✓ | ✓ | X |
| Quote Management | X | ✓ | ✓ |
| Real Time Notifications | ✓ | X | ✓ |
| Reporting & Statistics | ✓ | X | ✓ |
| Role-Based Permissions | ✓ | ✓ | X |
| Rules-Based Workflow | ✓ | ✓ | X |
| Sales Analytics | ✓ | ✓ | ✓ |
| Sales Forecasting | ✓ | X | ✓ |
| Sales Reporting | ✓ | X | ✓ |
| Sales Tracking | ✓ | ✓ | ✓ |
| Search Functionality | ✓ | ✓ | X |
| Social Media Integration | ✓ | X | ✓ |
| Territory Management | ✓ | ✓ | ✓ |
| Web Forms | ✓ | ✓ | X |
| Workflow Management | X | ✓ | ✓ |
| **Integration** |  |  |  |
| Constant Contact | ✓ | X | ✓ |
| Cyfe | ✓ | X | ✓ |
| Facebook | ✓ | X | X |
| G Suite | ✓ | ✓ | X |
| Gmail | ✓ | ✓ | ✓ |
| Google Analytics | ✓ | X | X |
| Google Calendar | X | ✓ | X |
| MailChimp | ✓ | X | ✓ |
| Microsoft Outlook | ✓ | ✓ | X |
| PayPal | X | ✓ | X |
| QuickBooks | X | ✓ | X |
| QuickBooks Online | ✓ | X | ✓ |
| Salesforce Sales Cloud | X | X | - |
| Slack | ✓ | X | X |
| Stripe | X | ✓ | X |
| Twitter | ✓ | X | X |
| Wordpress | ✓ | ✓ | X |
| Xero | X | X | X |
| Zapier | ✓ | X | ✓ |
| Zendesk | ✓ | X | ✓ |
| **Security** |  |  |  |
| Cloud Secuirty Alliance (CSA) Cloud Controls Matrix (CCM) | X | X | ✓ |
| Federal Risk and Authorization Management Program (FedRAMP) | X | X | ✓ |
| HIPAA Business Associate Agreement (BAA) | X | X | ✓ |
| ISO 27001/27002 | ✓ | X | ✓ |
| Payment Card Industry (PCI) Data Security Standards (DSS) | X | X | ✓ |
| SOC 1/SSAE 16/ISAE 3402 | X | ✓ | ✓ |
| SOC 2 | ✓ | X | ✓ |
| Zoho is SOC 2 Type II compliant and ISO 27001:2013 certified. [www.zoho.com](http://www.zoho.com) has datacenters are in USA. Those datacentre colocation facilities are SOC1 Type II certified. | ✓ | X | X |
| **Encryption** |  |  |  |
| Encryption of sensitive data at rest | X | ✓ | ✓ |
| HTTPS for all pages | ✓ | ✓ | ✓ |
| **Access Control** |  |  |  |
| Multi-factor authentication options | ✓ | X | ✓ |
| **Data Policy** |  |  |  |
| Data backup in multiple location/GEO regions | ✓ | ✓ | ✓ |

Table - Camparison of 3 most popular CRM systems

# New Ideas

The purpose of this project is to improve current CRM systems by creating a new CRM system on a software. Following on from the idea of improving current CRM systems, several ideas are proposed, but due to the time constraints of the project, not all the ideas that have been mentioned may be implemented into the system depending on how much time is spent on each idea takes to fully design and implement.

The ideas of this project are to create easier functionality and implement features that are the same as other CRM’s systems, with storing data to the cloud and more automation. This is to improve on current systems and to help people who use CRM systems speed up their processes.

The two ideas, storing data to the cloud and more automation, take current solutions that have already been done and change it to be suitable for my project, but to complete these ideas, tutorials will have to be followed as not enough personal knowledge has been learned from previous years.

In this section, it will detail the plans that have been created to create the prototype. This includes system flow diagram, tools required, requirement specification, features that will be included, a layout design and data dictionary for the tables in the database.

## Proposed System

The proposed system will be required to go through some changes but will still be attempting to provide a solution to the described problems by implementing the new ideas. The system would be required to perform the following functionality in order to meet the requirements:

1. **Automatically store data into the cloud and into the local file simultaneously**

The system should be coded to store the data to the local database file and into the cloud. This would mean that a database file would be in the cloud and in the local file already. With this, there is a backup database that is current and up-to-date locally and in the cloud, so that there will be no data loss if database had been deleted locally.

1. **Create, Read, Update, Delete (CRUD)**

The CRM system will be able to read the data from the database and output the correct table of data from the database to the users, whereas the users of the CRM system will be able to create a new record of data, update a current record of data and delete a record of data from the database whilst using the system.

1. **Customer Segmentation**

The user’s will be able to separate customers apart based on factors such as demographic, geographic, psychological and behavioural factors. This would be when the companies want to concentrate their marketing for certain customers and gain a competitive advantage by focusing on that specific segment compared to other companies.

1. **Data Search**

The system will be able to filter out records that do not match what the user is currently looking for by inputting what they are searching for and display the records that match or are similar to what they have searched.

1. **Email**

The user will allow to send emails from the system, whilst also being able to check their inbox and the information stored in that email.

## Data Flow Diagram

In the data flow diagram shown in Figure 7 shows the flow of the projected CRM system from the user to the system, to the local directory and to the cloud, in which all are connected to one another. A lot of company CRM’s store data directly to the cloud and removing files locally, whereas this project will do both. In an organisation, the user’s will be the employees, managers, directors etc. as they will be using the CRM system. These users will be inputting data such as contact details, inventory etc. into the system to input to the database and then data from the database will be loaded to the CRM system and displayed back towards the user to show them what’s in the database. The user will be able to sync the database to the cloud which in this project would be google drive. Whenever the system is loaded again, then the database that has been synced to the cloud, it will automatically download to the local folder for the system to read. This is so that the users who work outside work premises will be able to create, read, update and delete data from the database. Since it is uploaded into the cloud, the risk of losing information permanently is minimal because of having the database in the cloud and local directories on computers.

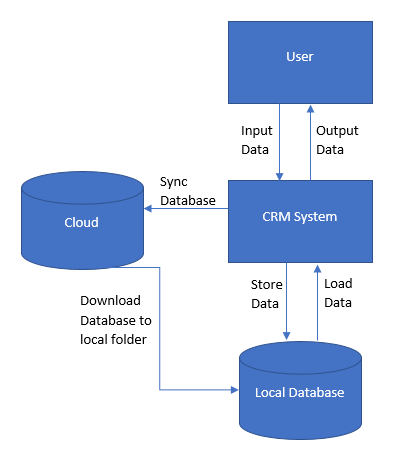


Figure - System Flow Diagram

## Tools

The main tools and skills used to develop this project is Microsoft Visual Studio, C#, Microsoft Access, Outlook and Google drive.

Microsoft Visual Studio is used to develop computer applications, web sites, web apps, web services and mobile applications. For this project, windows forms will be the development platform used on visual studio. It contains an integrated debugger that debugs at source level and at a machine level. In this project, visual studio allows a form designer to build a GUI application, which allows easier designing during the design phase. Visual studio supports several programming languages and can debug them, which range from C, C++, Visual Basic .NET, C#, Javascript, HTML, CSS etc.

C# is a simple, general-purpose, objected oriented programming language and is like C and C++. C# is easy to learn, can be compiled on a variety of computer platforms and is part of .Net framework. It provides important features, such as Boolean conditions, automatic garbage collection, properties and events, simple multithreading and much more.

Outlook application would be used as it is mainly used as an emailing application and can is compatible with C#. It will be implemented to the system to view an inbox and sending emails.

One of the most important tools for this project, is the database management system (DBMS) as one of the main functions is to create, read, update and delete data from a database and in which the DBMS used is Microsoft Access as it can store data in its own format, but more importantly it can import, or link data stored directly to other applications and databases. Reason for use is because it is supported by visual studio applications, so that the data within Access can be added, read, updated and deleted, also it will be easier to upload the database to the cloud and download from the cloud.

Google drive is an essential part of this project, as it will be used to store the database into a cloud. Therefore, Google API will be required and google developers will be needed to integrate the CRM system.

## Features compared to other companies

The project’s CRM system’s proposed features are compared to other business as shown in table 2. These companies sell the same type of product, but have different features implemented. Salesforce does not suit freelancers as they are expensive and are mainly used in businesses which is a downfall. Apptivo is the only company to use google drive, even though it is a free storage where documents can be shared and could be used as back up storage, but they Zoho and Salesforce have not implemented that feature as they directly work on the cloud. The 3 company CRM systems do not have all the features ticked but prove to be so successful still.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | **My CRM** | **Zoho CRM** | **Apptivo CRM** | **Salesforce CRM** |
| **Platform Supported** |  |  |  |  |
| Cloud based | ✓ | ✓ | ✓ | ✓ |
| **Typical Customers** |  |  |  |  |
| Freelancers | ✓ | ✓ | ✓ | X |
| Small businesses | ✓ | ✓ | ✓ | ✓ |
| Mid-size Business | ✓ | ✓ | ✓ | ✓ |
| Enterprise | ✓ | ✓ | ✓ | ✓ |
| **Features** |  |  |  |  |
| API | ✓ | ✓ | ✓ | ✓ |
| Activity Dashboard | ✓ | ✓ | ✓ | ✓ |
| Activity Tracking | X | ✓ | X | ✓ |
| Ad hoc Reporting | X | ✓ | X | ✓ |
| Approval Process Control | X | ✓ | X | ✓ |
| Auto-Responders | X | ✓ | ✓ | X |
| Automatic Reminders | ✓ | X | ✓ | ✓ |
| Calendar Management | ✓ | ✓ | ✓ | X |
| Calendar Sync with Google | ✓ | ✓ | ✓ | X |
| Campaign Management | ✓ | ✓ | X | ✓ |
| Configurable Workflow | ✓ | ✓ | ✓ | ✓ |
| Contact Database | ✓ | ✓ | X | ✓ |
| Contact Management | ✓ | X | ✓ | ✓ |
| Customer History | ✓ | ✓ | ✓ | ✓ |
| Data Visualization | ✓ | ✓ | X | ✓ |
| Document Storage | X | ✓ | ✓ | X |
| Email Integration | ✓ | ✓ | ✓ | X |
| Email Templates | ✓ | ✓ | ✓ | ✓ |
| Filtered Views | ✓ | ✓ | ✓ | ✓ |
| Use of Google Drive Storage | ✓ | X | ✓ | X |
| Inbox Management | ✓ | ✓ | X | ✓ |
| Lead Assignment | X | ✓ | ✓ | X |
| Lead Capture | X | ✓ | ✓ | X |
| Lead Distribution | ✓ | ✓ | ✓ | ✓ |
| Lead Generation | X | ✓ | ✓ | X |
| Lead Management | ✓ | ✓ | ✓ | ✓ |
| Lead Scoring | ✓ | ✓ | ✓ | ✓ |
| Microsoft Outlook Integration | ✓ | ✓ | X | ✓ |
| Multi-Currency | X | ✓ | ✓ | ✓ |
| Opportunity Management | ✓ | ✓ | ✓ | ✓ |
| Pipeline Management | ✓ | ✓ | ✓ | ✓ |
| Pipeline Reports | ✓ | ✓ | ✓ | ✓ |
| Purchase Order Management | ✓ | ✓ | ✓ | X |
| Quote Management | X | X | ✓ | ✓ |
| Real Time Notifications | X | ✓ | X | ✓ |
| Reporting & Statistics | X | ✓ | X | ✓ |
| Role-Based Permissions | ✓ | ✓ | ✓ | X |
| Rules-Based Workflow | ✓ | ✓ | ✓ | X |
| Sales Analytics | ✓ | ✓ | ✓ | ✓ |
| Sales Forecasting | X | ✓ | X | ✓ |
| Sales Reporting | X | ✓ | X | ✓ |
| Sales Tracking | ✓ | ✓ | ✓ | ✓ |
| Search Functionality | ✓ | ✓ | ✓ | X |
| Social Media Integration | ✓ | ✓ | X | ✓ |
| Web Forms | X | ✓ | ✓ | X |
| Workflow Management | X | X | ✓ | ✓ |
| **Integration** |  |  |  |  |
| Facebook | ✓ | ✓ | X | X |
| G Suite | X | ✓ | ✓ | X |
| Gmail | ✓ | ✓ | ✓ | ✓ |
| Google Analytics | X | ✓ | X | X |
| Google Calendar | ✓ | X | ✓ | X |
| Microsoft Outlook | ✓ | ✓ | ✓ | X |
| PayPal | X | X | ✓ | X |
| Twitter | X | ✓ | X | X |

Table - My CRM compared to Others

## Proposed Layout Design

The following layout designs are planned and not fully finalised as some changes will be made during the creation of the prototype. It is to give a rough idea of what each design is going to look like.

### Dashboard

This is the proposed dashboard layout shown in Figure 8, which is the main page that is first seen by the user when the system loads. It has charts representing a list of data such as sales, customer service etc. The dropdown menu is used to allow the user to change what data they want to view in the chart.

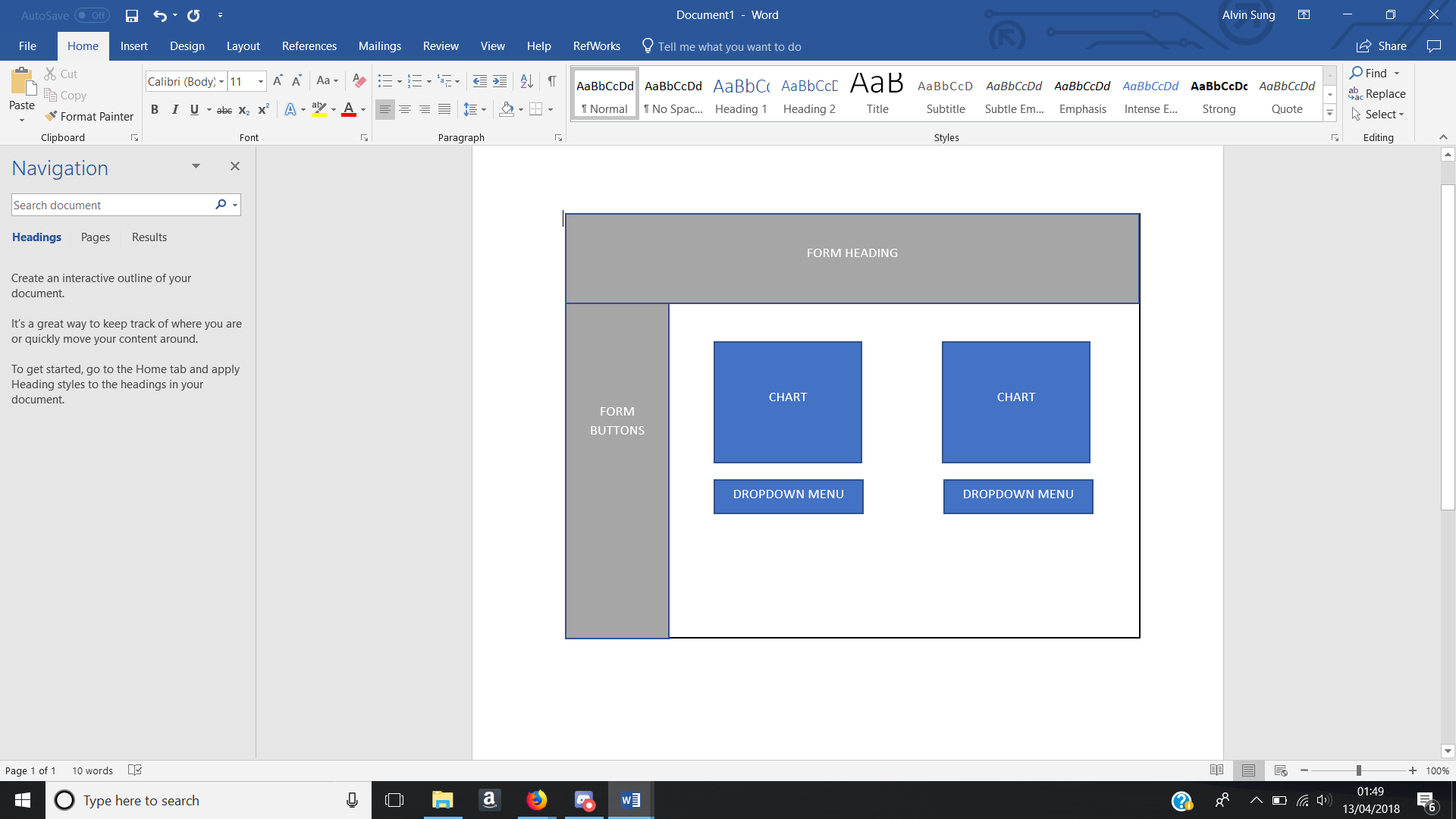


Figure - Proposed Dashboard Layout

### Contacts Form

The planned implementation of the contacts form is shown in figure 9. It has multiple textboxes that will require data to be inputted by the user and an image box with a button underneath which will allow the user to select an image for that contact. At the very bottom of the form will have four buttons which will allow data to be created, updated and deleted to the database and a data grid view which will read from the database.



Figure - Proposed Contacts Form Layout

### Calendar Form

The calendar form planned layout has two textboxes, a date picker and a time picker as shown in figure 13, which will require the user to input data into the textboxes and select a date and time which will be inputted into the database. Underneath the form controls will have four buttons which will allow data to be created, updated and deleted to the database and a calendar which will read from the database visually showing what event they have on each day.

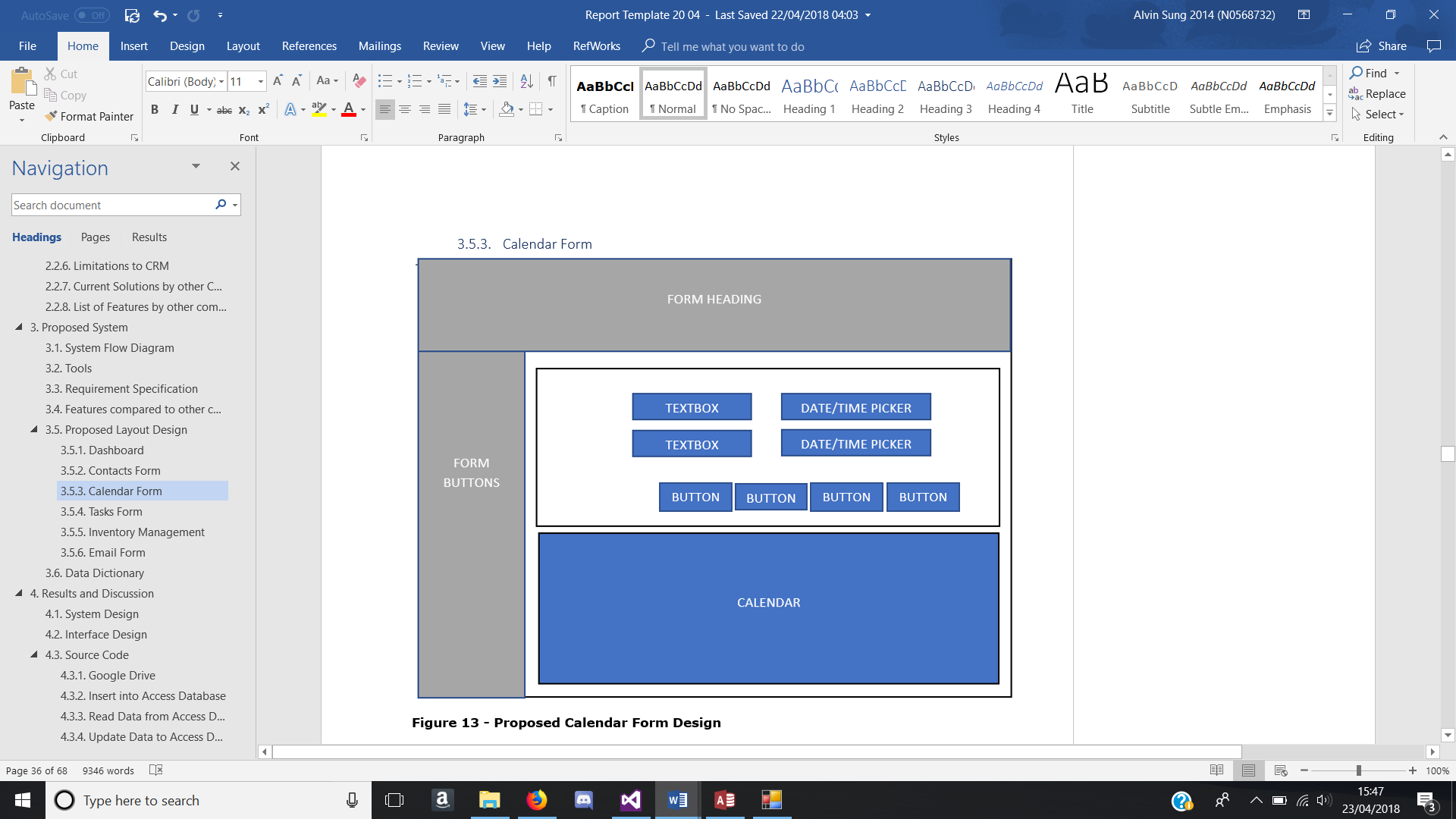


Figure – Proposed calendar form layout

### Tasks Form

The proposed calendar form layout has 5 types of form controls, which consist of multiple textboxes, a date picker and a time picker which will require the user to input data into the textboxes and select a date and time which will be inputted into the database. Underneath them form controls, there are four buttons which will allow data to be created, updated and deleted to the database and a datagridview which will read all data from the database and a calendar which will filter specific tasks on the selected date.

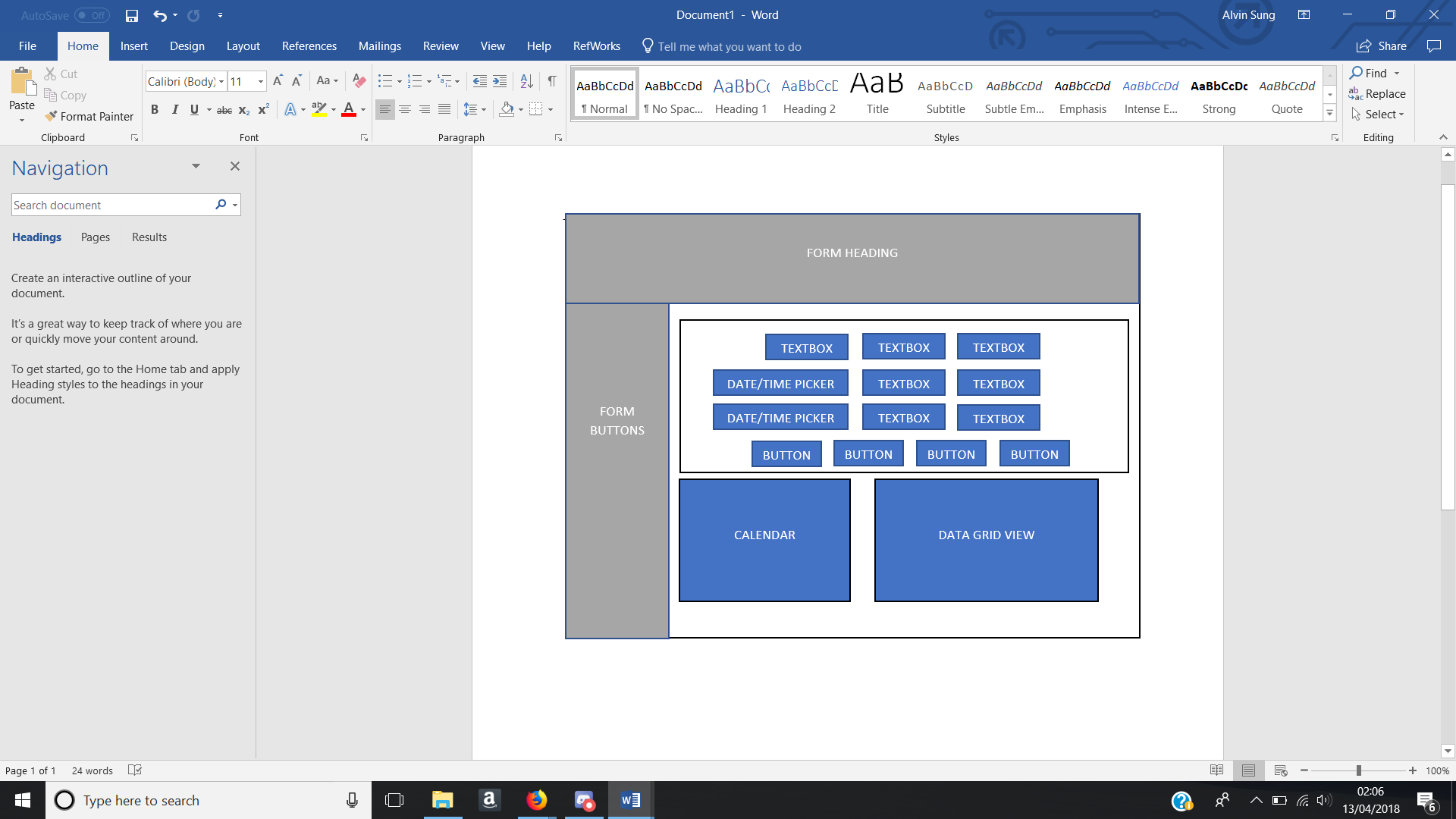


Figure - Proposed tasks form layout

### Inventory Management

This is the proposed inventory management form shown in figure 15. It has multiple textboxes that will require data to be inputted by the user and dropdown boxes must have a selected value. At the very bottom of the form will have four buttons which will allow data to be created, updated and deleted to the database and a datagridview which will read from the database.

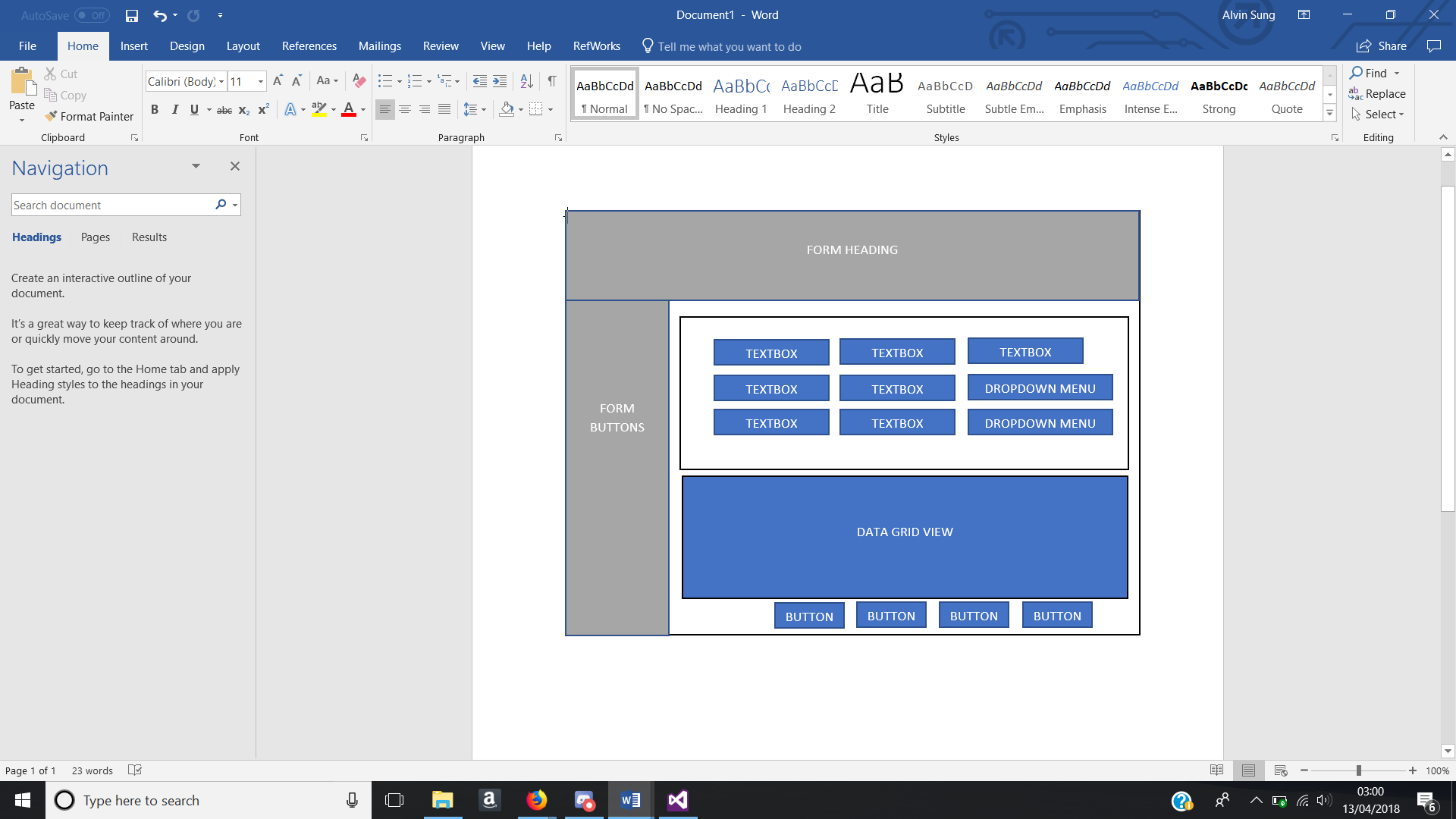


Figure - Proposed Inventory Management Form

### Email Form

This is the proposed email form shown in figure 16. It has multiple textboxes that will require data to be inputted by the user and dropdown boxes must have a selected value. There is a button that will be used to attach a file and a send and refresh button at the bottom of the form. The data grid view will be used to show emails inbox and the refresh button will be used to refresh the data grid view.

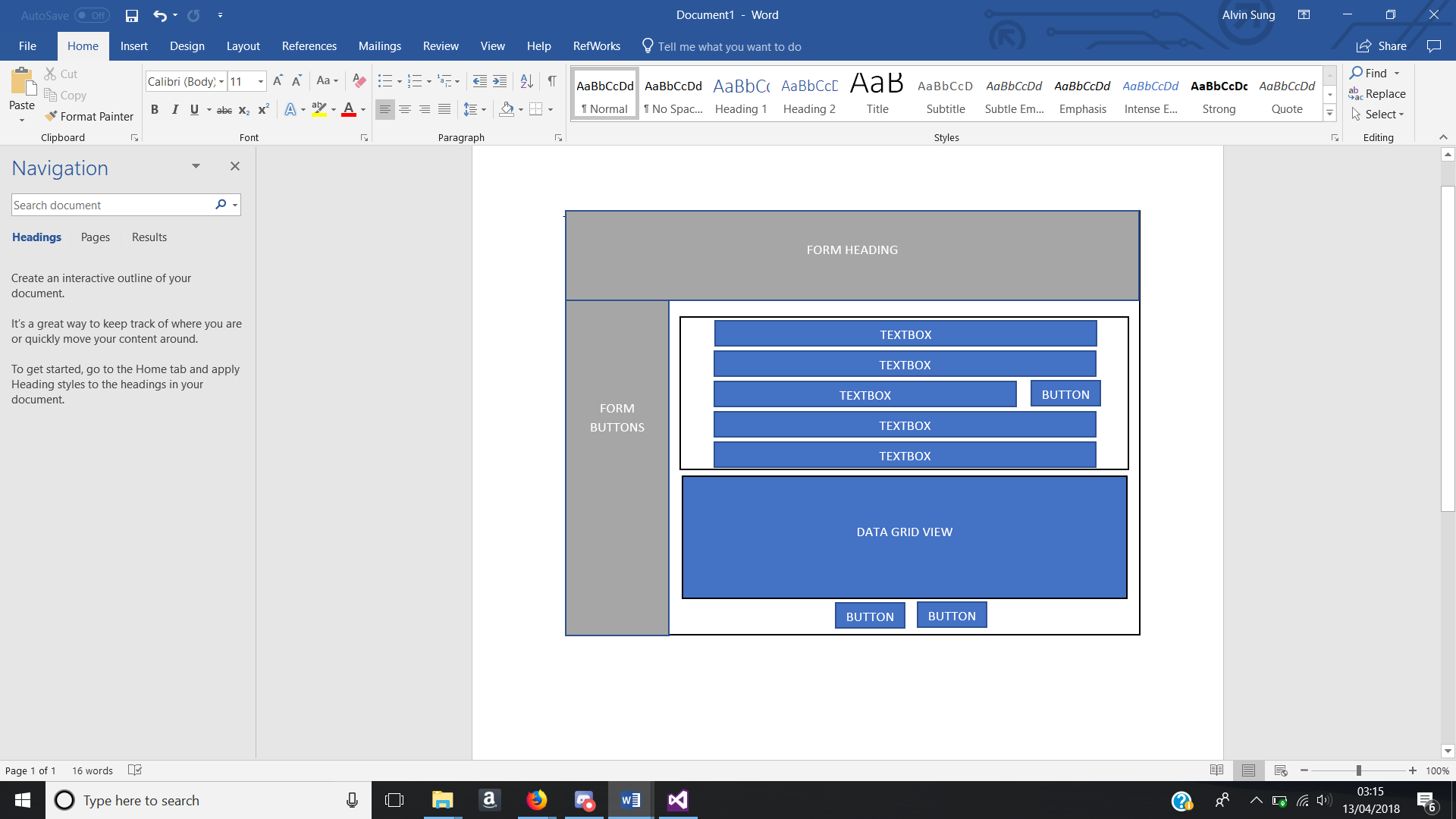


Figure - Proposed email form layout

### Customer Form

The customer form shown in figure 17 will have a data grid view to output data for the user and a textbox that will be used to filter the data in the data grid view.

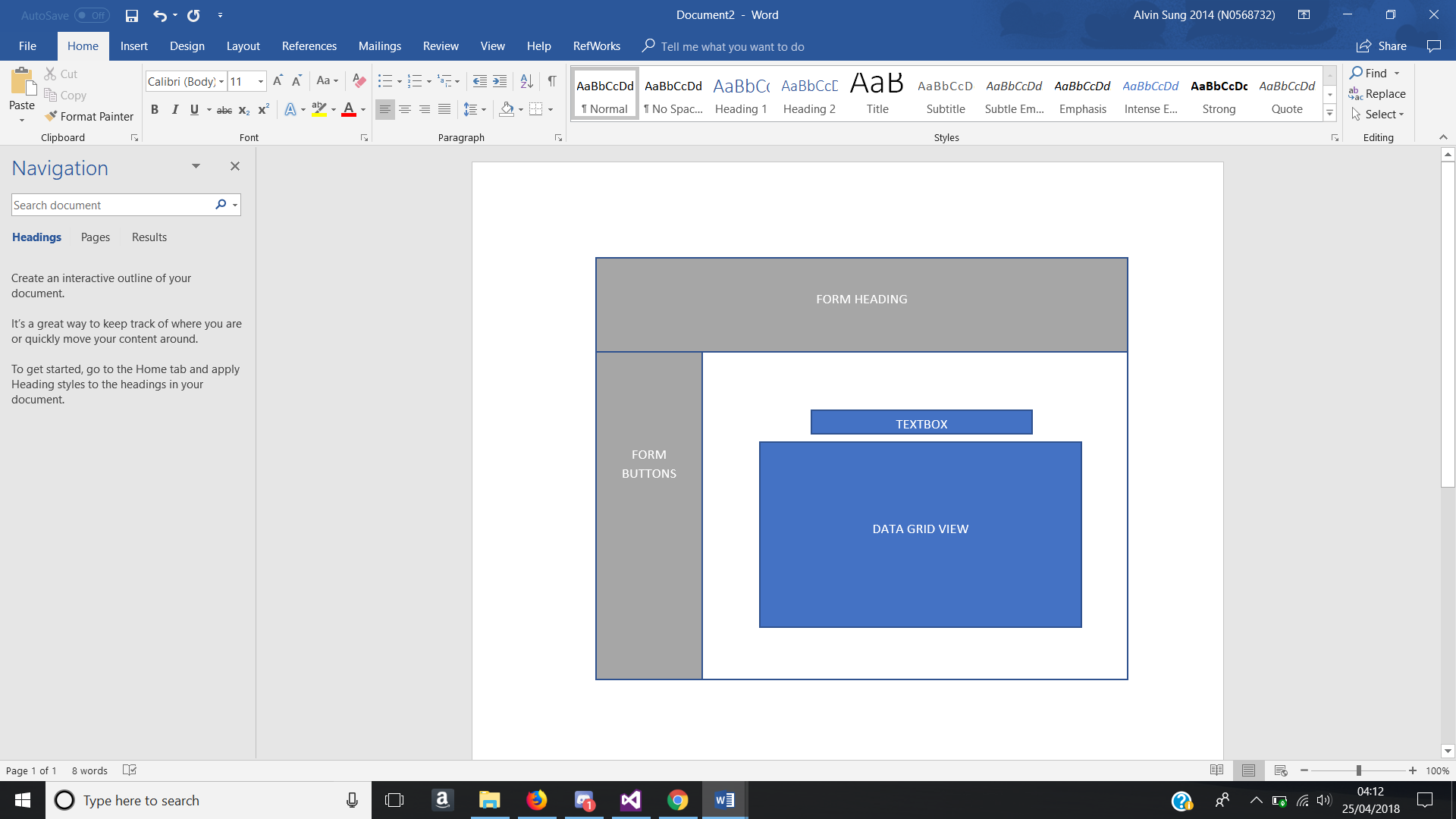


Figure - Proposed Customer Form Layout

## Data Dictionary

This section involves multiple tables that will be stored into the database. Each table includes the attributes, data types, description, nulls and format. Within the database the tables that will be stored are all the attributes in the contacts table shown in figure 3, events table shown in table 4, tasks table shown in table 5, inventory table shown in table 6, item sales table shown in table 7, monthly sales table shown in table 8 and the customer table shown in table 9. Every table will have data being inputted from the system excluding the customer table. The customer table will not have anything inputted from the system but will have pre-set data since in reality the customer table would be filled by customer signing into a company’s website.

**Contacts Table:** Store all contact information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Data Types** | **Description** | **Nulls** | **Field Length** |
| ID | AutoNumber | Unique number for each contact | No |  |
| Company | Text | Contact’s company name | No |  |
| Last Name | Text | Last name of contact | Yes |  |
| First Name | Text | First Name of contact | Yes |  |
| Email | Text | Contact’s Email Address | No |  |
| Job Title | Text | Current contacts job title | Yes |  |
| Business Phone | Number | Contact’s business phone number | Yes | 11 |
| Home Phone | Number | Contact’s home phone number | Yes | 11 |
| Mobile Phone | Number | Contact’s Mobile Number | Yes | 11 |
| Fax Number | Number | Contact’s Fax Number | Yes |  |
| Address | Text | Where the contact lives | Yes |  |
| City | Text | The city the contact lives in | No |  |
| State | Text | The state the contact lives in | Yes |  |
| Postcode | Text | Contact’s postcode | Yes | 7 |
| Country | Text | Country the contact lives in | No |  |
| Webpage | Text | Webpage that the contact has | Yes |  |
| Notes | Text | Description/Extra information of contact | Yes |  |

Table - Contacts Database Data

Events Table: Store information of all events

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Data Types** | **Description** | **Nulls** | **Field Length** |
| EventID | AutoNumber | Unique identifier | No |  |
| EventName | Text | Name of the event | No |  |
| EventDesc | Text | Description of the event | Yes |  |
| EventDate | Date | Date of the event | No |  |
| Duration | Text | How long the event will be | No |  |

Table - Events Database Data

**Tasks Table:** Store information of all Tasks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Data Types** | **Description** | **Nulls** | **Field Length** |
| TaskID | AutoNumber | Unique identifier | No |  |
| TaskName | Text | Name of the trask | No |  |
| TaskDesc | Text | Description of the task | Yes |  |
| StartDate | Date | Date of task when started | No |  |
| DueDate | Text | Date of task when due. | No |  |

Table - Tasks Database Data

**Inventory Table:** Store information of inventory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Data Types** | **Description** | **Nulls** | **Field Length** |
| SerialNo | Text | Unique Identifier | No |  |
| Item Name | Text | Name of the item | No |  |
| Client Company | Text | Company that selling item | No |  |
| Order ID | Text | Unique Identifier | No |  |
| Date Order | Text | Date when item ordered | No |  |
| Date Shipped | Text | Date when shipped to company | No |  |
| Model No | Text | Model number | No |  |
| Type | Text | Type of item | No |  |
| Maintenance Date | Date | Date of maintenance | No |  |
| Price | Number | Price of the item | No |  |

Table - Inventory Database Data

**Item Sales Table:** Store information of item sales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Data Types** | **Description** | **Nulls** | **Field Length** |
| ItemID | AutoNumber | Unique Identifier | No |  |
| Item Name | Text | Name of the Item | No |  |
| Sold | Number | How many of that specific item has been sold | No |  |

Table – Item Sales Database Data

**Monthly Sales Table:** Store information of total sales every month

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Data Types** | **Description** | **Nulls** | **Field Length** |
| ID | AutoNumber | Unique Identifier | No |  |
| Month | Text | Months in the year | No |  |
| Total Sales | Number | Total amount of sales | No |  |

Table - Monthly Sales Database Data

**Customer Table:** Store all information of customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Data Types** | **Description** | **Nulls** | **Field Length** |
| CustomerID | AutoNumber | Unique Identifier | No |  |
| First Name | Text | First name of the customer | No |  |
| Surname | Text | Surname of the customer | No |  |
| Address | Text | The street address where the customer lives | No |  |
| City | Text | The city the customer lives in | No |  |
| Postcode | Text | Customer’s postcode | No | 7 |
| E-mail | Text | Customer’s email | No |  |
| Phone Number | Integer | Customer’s phone number | No | 11 |

Table - Customer Database Data

# Implementation

This section of the report focuses on the actual creation of the system. The methodology used throughout the project was an Incremental Build Model, which is a method that relies on repetitive implementation and testing on individual features before moving on. The incremental build model methodology would be suitable for the development of this system because it would allow to see what can and cannot be done whilst going through the development as the requirements and designs would change. All aspects of the prototype would have to be tested out because most of the features of the project, such as uploading data to the cloud is not working correctly, this would not allow the system to be used on another computer, as the database would only be stored in a local folder. The reason for not using the Waterfall methodology is because it would require coding the entire system and then test the system. The problem with that, is that multiple errors will be discovered and fixing them all will be difficult and long considering the time restraint that this project has.

This CRM project is using C# as the main developing language, it is a system made in windows form on visual studio, which will store in 1 folder. The folder is named CRM Project and inside the folder contains the project.snl (Microsoft Visual Studio Solution) file and 3 folders, which are .vs folder, images folder and CRM folder. Within the CRM folder contains a bin, obj and properties folder, an App XML config file, all form code and form designer code in C# and the project file. The bin folder contains 2 folders called Debug and Release, and within the Debug folder, contains database that is used in the system, Application file of the system, a .exe file and other type of files that should not be tampered with.

## Interface Design

**Header Design**

As shown in figure 18, The header design is at the top of each form with a heading of the current form to inform the user what form they are currently on. A logo is at the far left which will also be above the navigation panel.

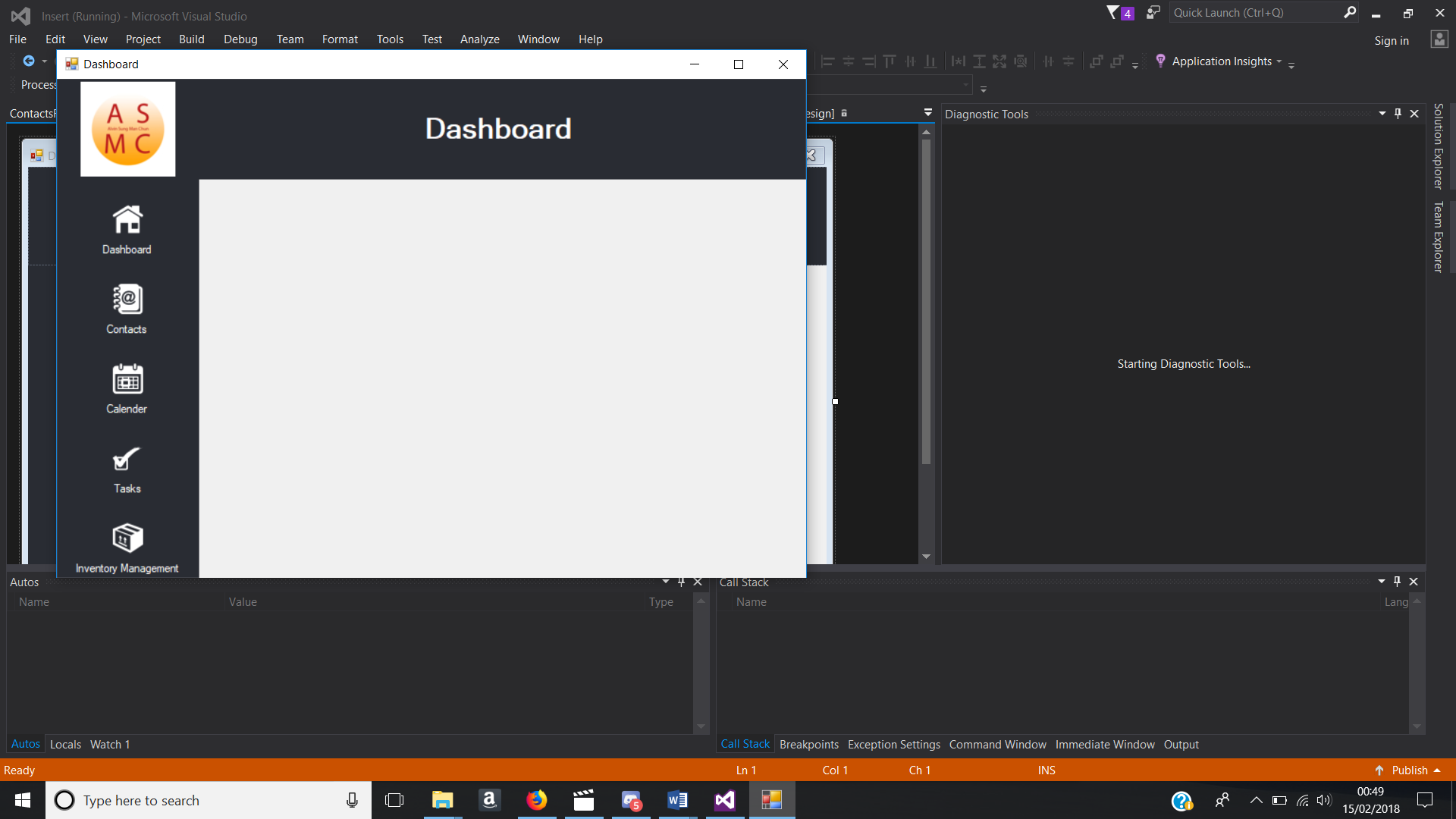


Figure - Header Design

**Navigation Design**

The navigation is to allow the user to access different forms of the system. It is designed to be on the left side of every single form with the exact same design. The design is made with a panel control and buttons within that panel as shown in figure 19. The buttons are designed to be flat, so it looks integrated with the panel and has an image and text for each button to inform the user what button they are associated with.

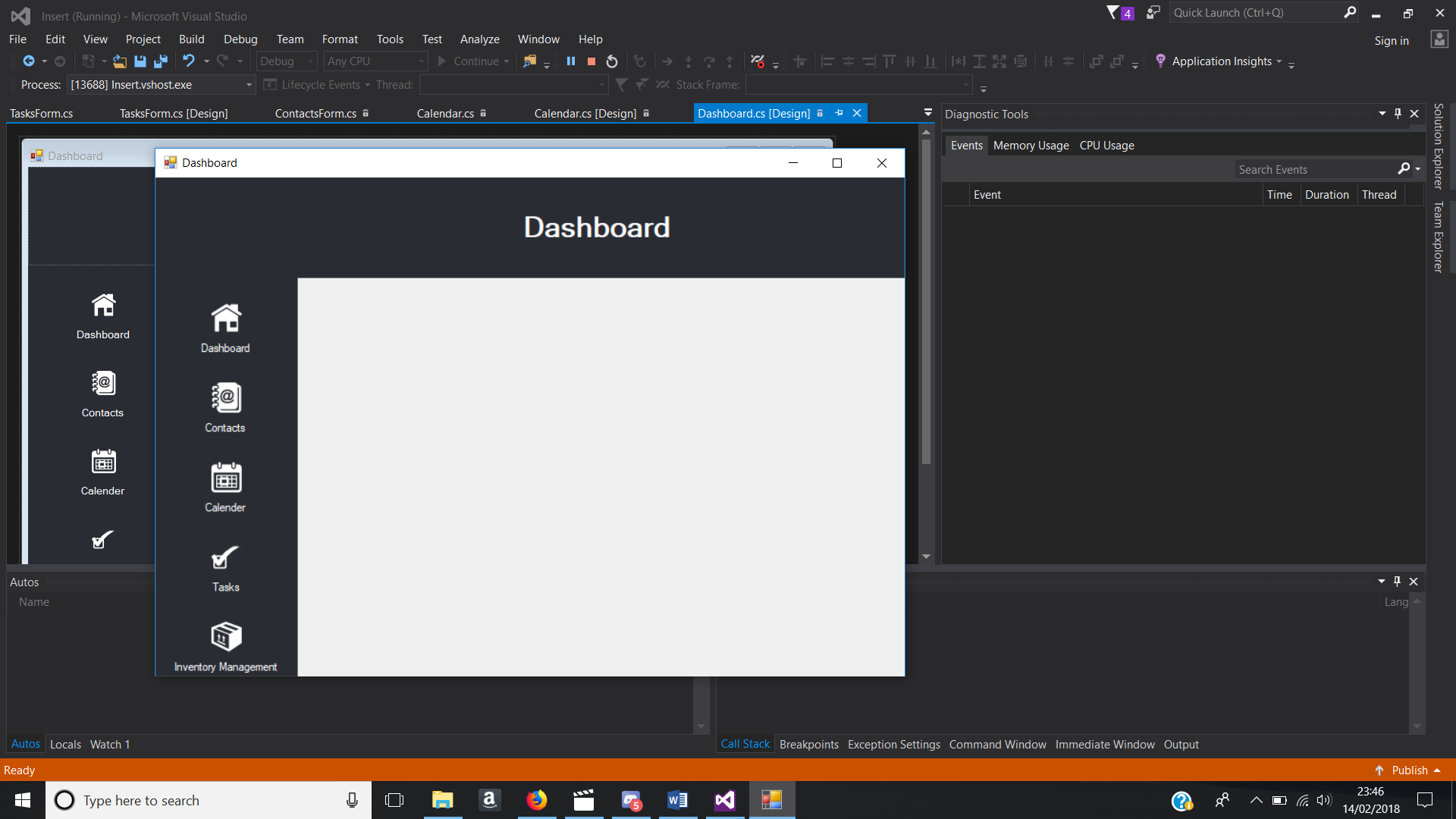


Figure - Navigation Design

**Dashboard Form**

The dashboard form as shown in figure 20, is the main page that is first seen by the user when the system loads. It contains a chart which is to represent data depending on what the user selects in the dropdown menu. The dropdown menu is used to allow the user to change what data they want to view in the chart.

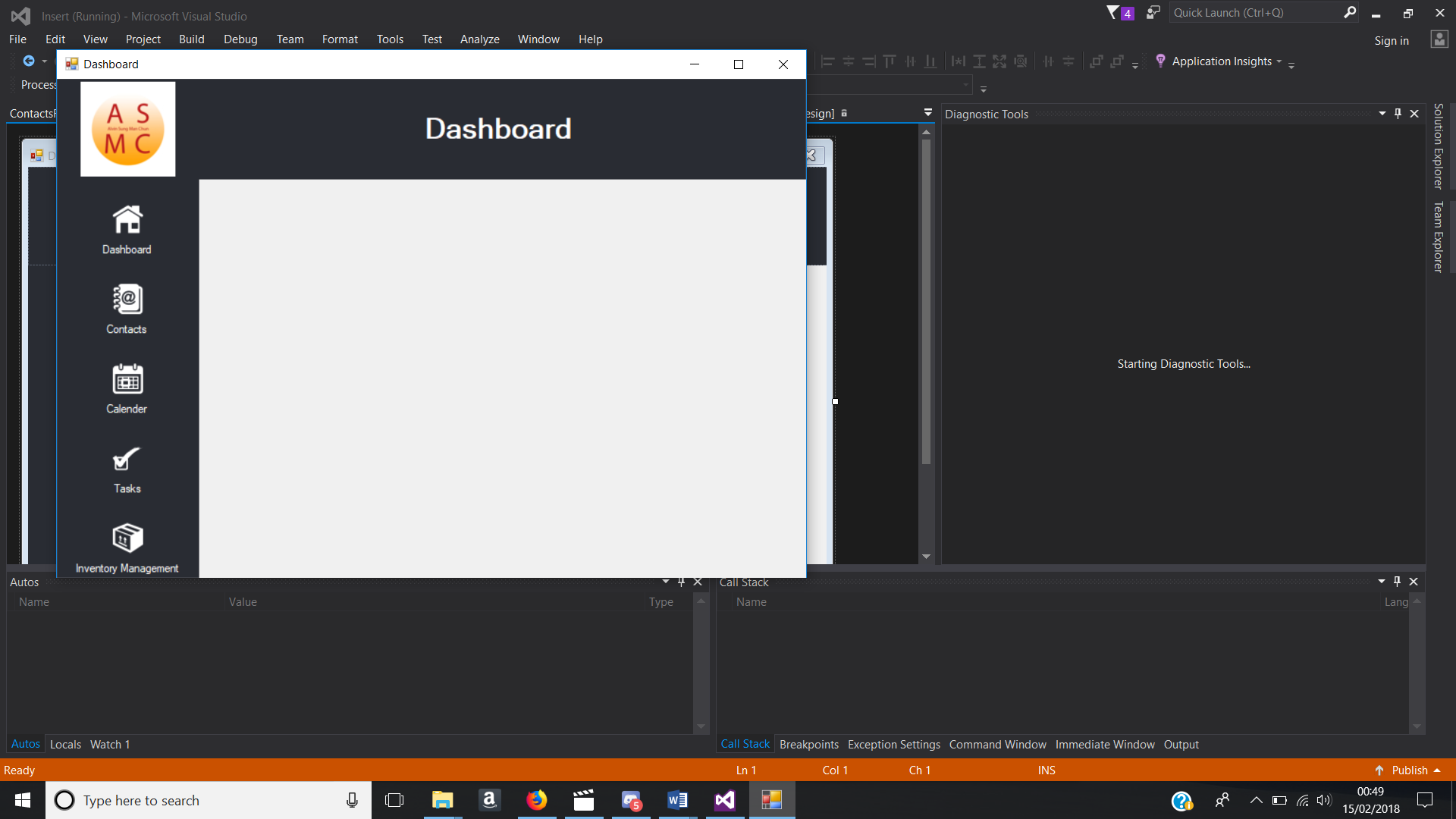


Figure - Dashboard Form

**Contacts Form**

The contacts form allows the user to add contacts to the database and view their contacts details. As shown in figure 21, it contains multiple text box which are disabled so that the user cannot input any data, a data grid view that shows data from the database and multiple buttons which do different functions.

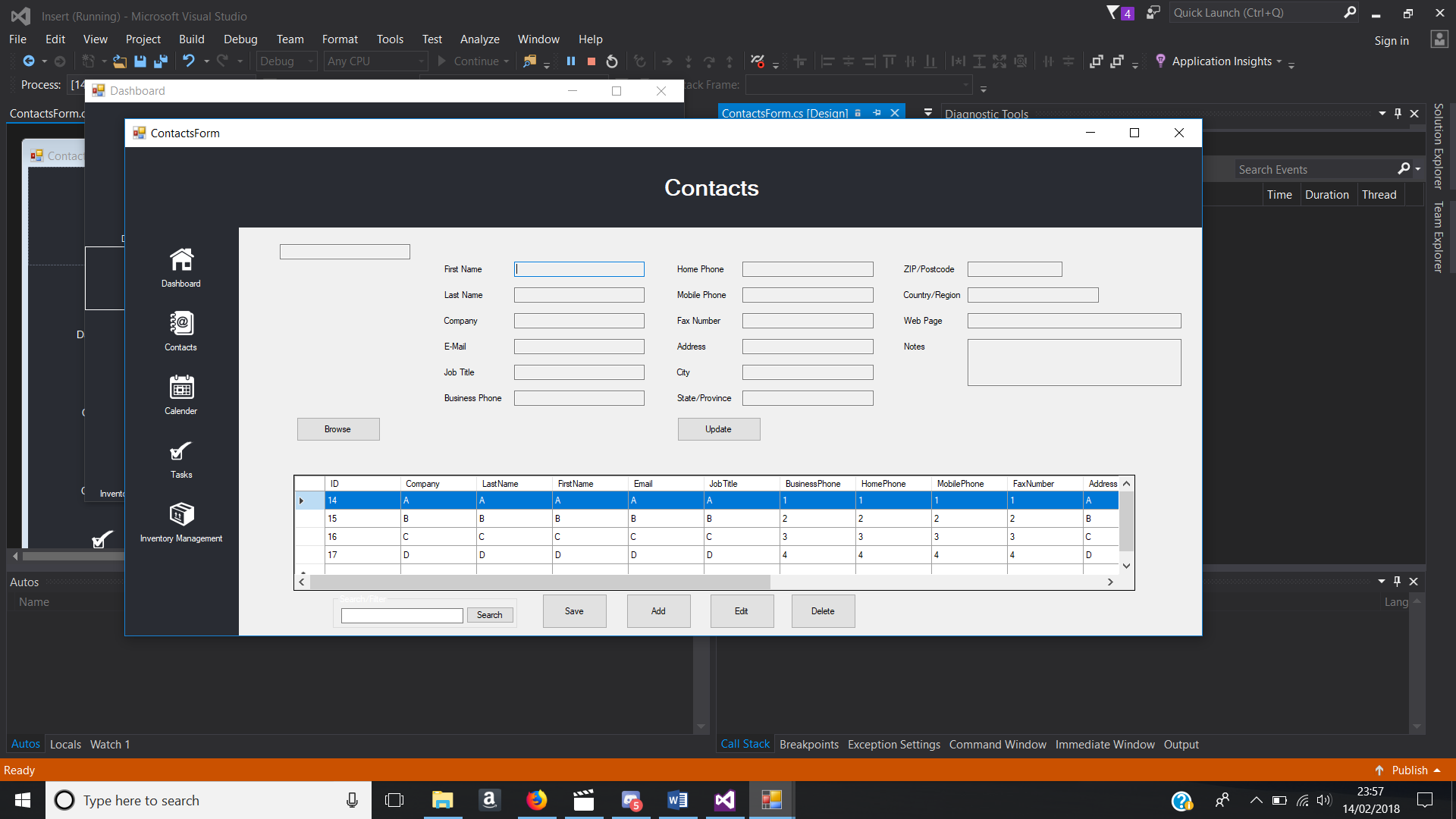


Figure - Contacts Form

**Calendar Form**

As shown in figure 22, the form has a month calendar for the user to select a date, a data grid view which shows data from the database, textboxes, a date picker, a dropdown menu and multiple buttons. The calendar form allows the user to input events such as meetings as a reminder for what they have for that specific day they select as well as seeing what events they have for the current day.

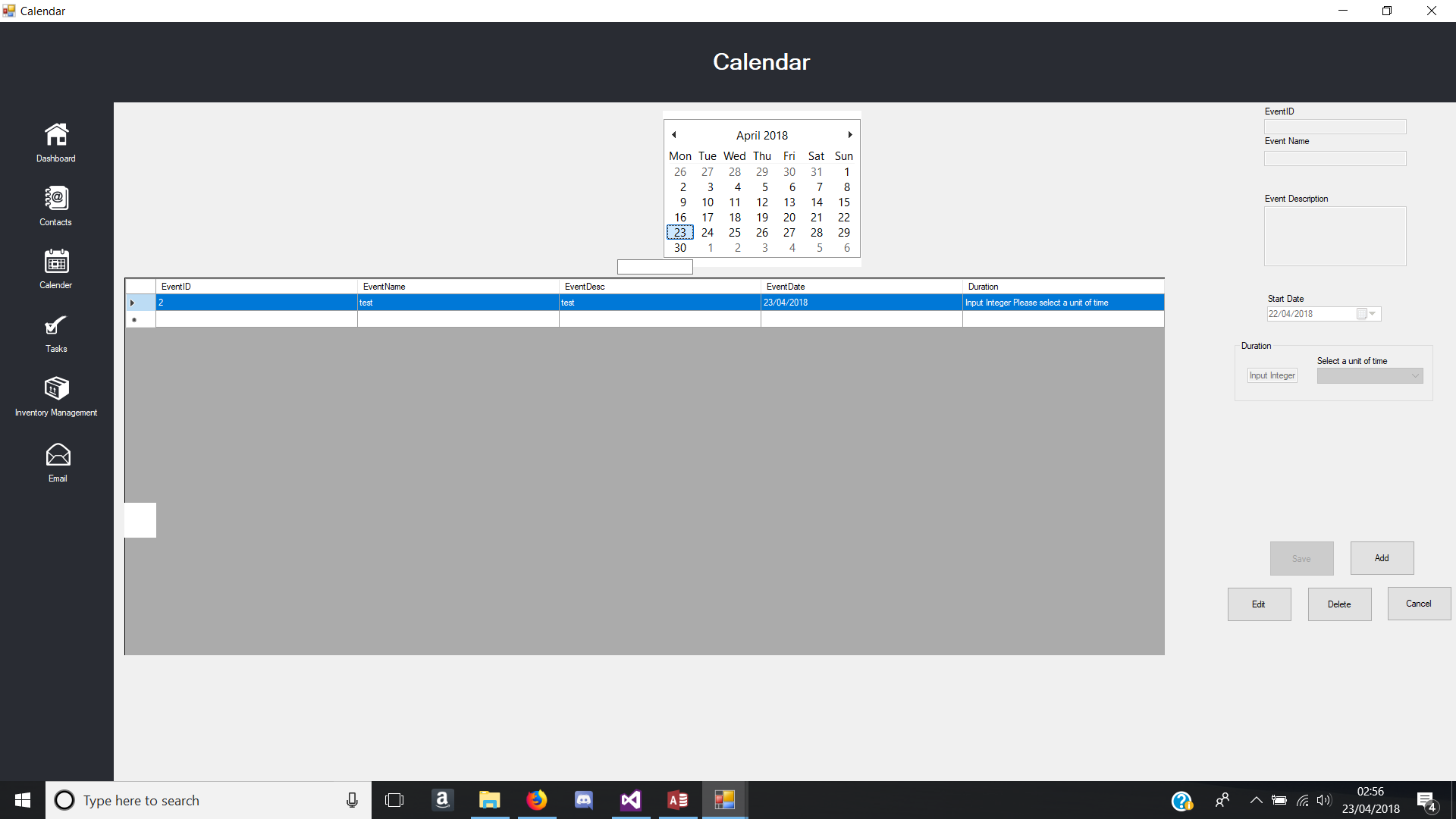


Figure - Calendar Form

**Tasks Form**

The tasks form is similar to the calendar form with the form controls as shown in figure 23. This form allows the user to input tasks that they need to do as a reminder for them, which will show current tasks that have been inputted too.

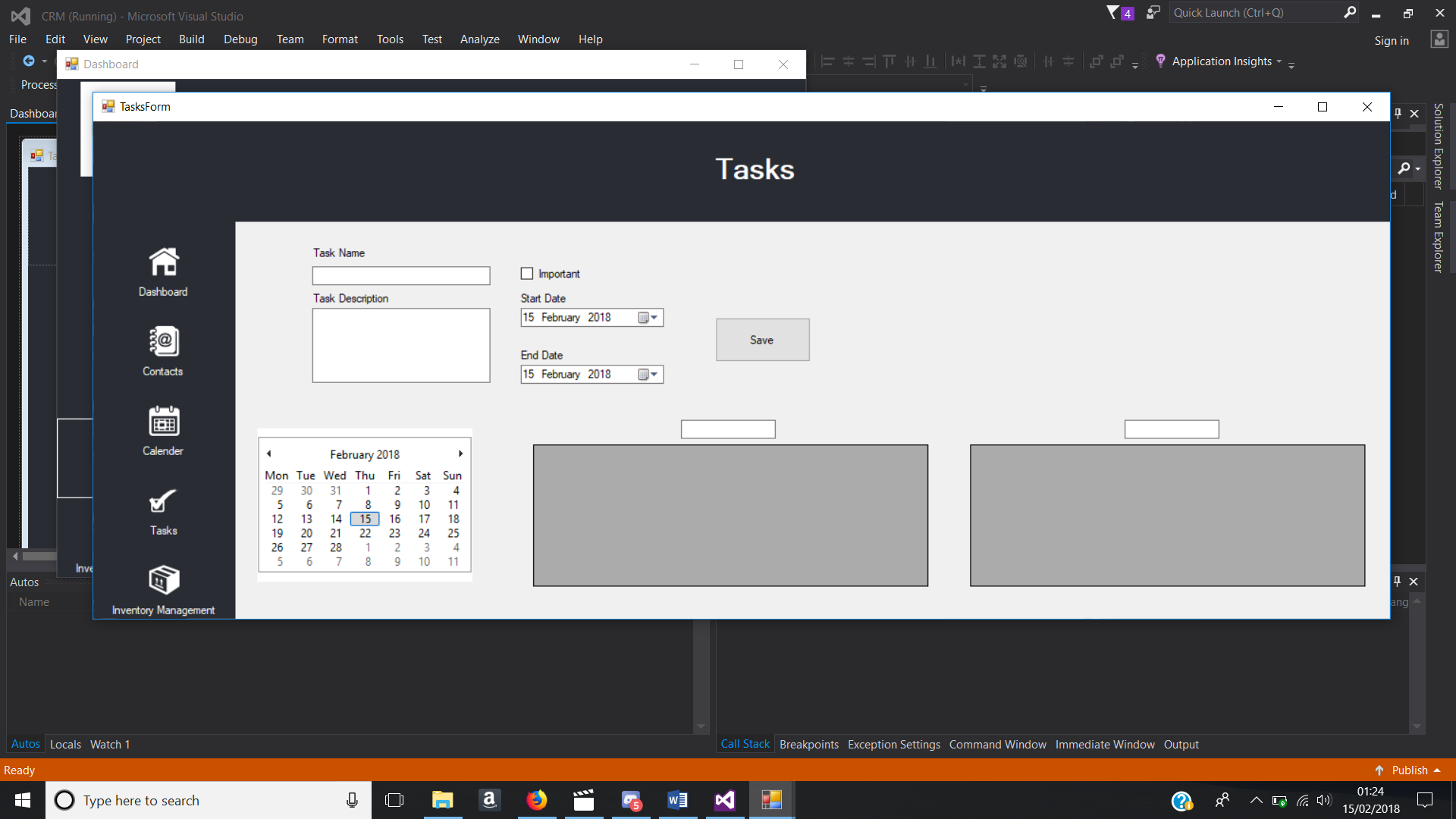


Figure - Tasks Form

**Inventory Form**

As shown in figure 24, this form shows the user the company inventory that is stored in the database. It has a data grid view which outputs all the data within the inventory table in the database, multiple textboxes for the user to input data and buttons that provide different functionality from enabling textboxes and dropdown menu to storing the data inputted into the database.

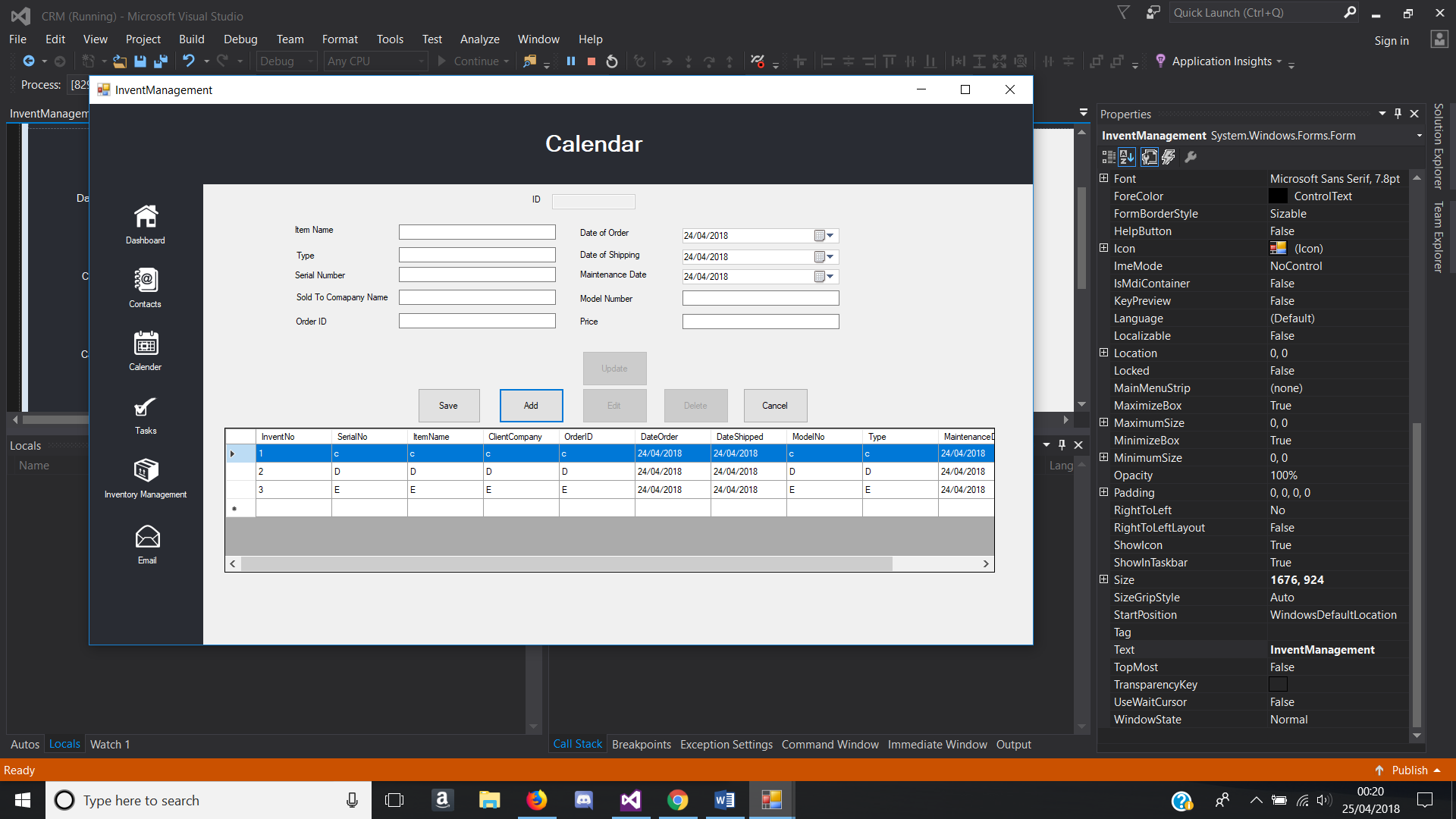


Figure - Inventory Form

**Email Form**

The email form contains textboxes that must require the user to input the correct data in, an attach button which opens up a file dialog to select a file to attach to the email, a send button which sends the email once all the textboxes are fully inputted and a refresh button which refreshes the data grid view. The data grid view is used as an inbox for emails and a web viewer is placed at the bottom right to read the body of the email as shown in figure 25.

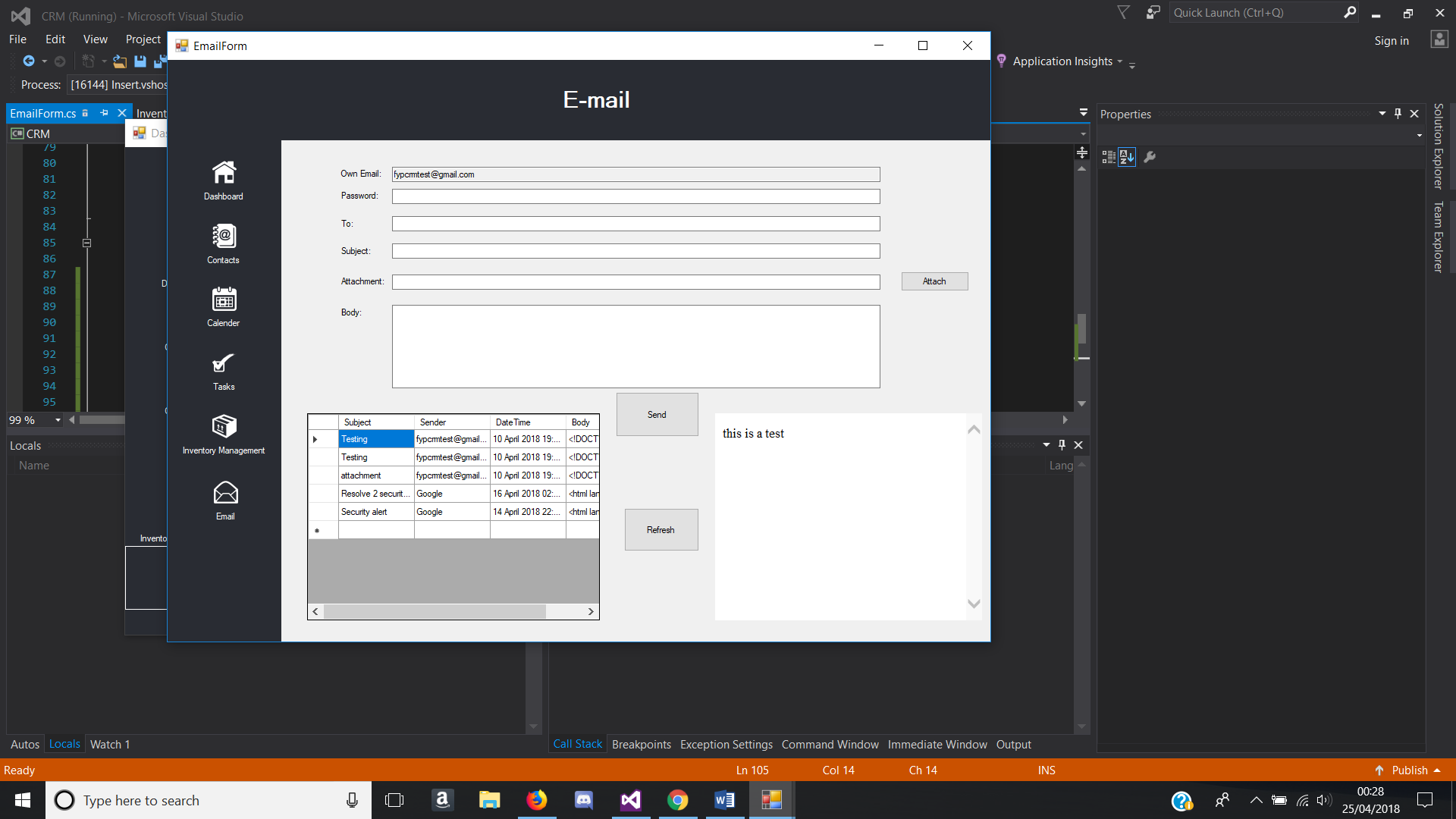


Figure - Email Form

## Source Code

The following list of source code is to show how the requirements were implemented in the system and what the code does.

### Chart Control

The source code shown below is to allow the chart to read from the database, which will then represent the data in a chart and when an item is selected in the dropdown list the chart will then change and read a different table in the database and represent that data into a chart.

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox1.Text == "Item Sales")

{

str3 = "SELECT Item,Sold from ItemSales";

com3 = new OleDbCommand(str3, cnon);

OleDbDataAdapter da = new OleDbDataAdapter(com3);

DataTable dt = new DataTable();

da.Fill(dt);

cnon.Close();

string[] N = new string[dt.Rows.Count];

int[] M = new int[dt.Rows.Count];

for (int i = 0; i < dt.Rows.Count; i++)

{

N[i] = dt.Rows[i][0].ToString();

M[i] = Convert.ToInt32(dt.Rows[i][1]);

}

chart1.Series[0].Points.DataBindXY(N, M);

}

else if (comboBox1.Text == "Monthly Sales")

{

str3 = "SELECT Months,TotalSales from Sales";

com3 = new OleDbCommand(str3, cnon);

OleDbDataAdapter da = new OleDbDataAdapter(com3);

DataTable dt = new DataTable();

da.Fill(dt);

cnon.Close();

string[] N = new string[dt.Rows.Count];

int[] M = new int[dt.Rows.Count];

for (int i = 0; i < dt.Rows.Count; i++)

{

N[i] = dt.Rows[i][0].ToString();

M[i] = Convert.ToInt32(dt.Rows[i][1]);

}

chart1.Series[0].Points.DataBindXY(N, M);

chart1.Series[0].LegendText = "Total Sales";

}

}

### Google Drive

Here is the source code for the implementation of google API to upload and download files from google drive. This required having 3 classes, CloudUpload, CloudDownload and MimeTypeConverter and these classes do not have any graphical user interface, instead it is just pure code which will be executed when called from by a form on load or form control such as a button. The MimeTypeConverter class is to convert the type of the file that will be uploaded such as a .doc (word document), .pptx (powerpoint) etc. The CloudUpload class is code which gets the file from the computer and uploads and edits that file into google drive. This would be called when the sync with google drive button is pressed. The CloudDownload class is used on the initialisation of the entire system, so when the system first loads, the file from google drive will automatically download into the targeted file which is the local directory of the system.

**MimeTypeConverter Class Code**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Insert

{

public class MimeTypeConvert

{

public string extension { get; set; }

public string mimeType { get; set; }

public string converterType { get; set; }

public MimeTypeConvert()

{

}

public MimeTypeConvert(string extension, string type, string converter)

{

this.extension = extension;

mimeType = type;

converterType = converter;

}

}

public static class MimeConverter

{

public static List<MimeTypeConvert> mimeList()

{

List<MimeTypeConvert> list = new List<MimeTypeConvert>();

list.Add(new MimeTypeConvert(".xlsx", "application/vnd.google-apps.spreadsheet", "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"));

list.Add(new MimeTypeConvert(".doc", "application/vnd.google-apps.document", "application/vnd.openxmlformats-officedocument.wordprocessingml.document"));

list.Add(new MimeTypeConvert(".pptx", "application/vnd.google-apps.presentation", "application/vnd.openxmlformats-officedocument.presentationml.presentation"));

list.Add(new MimeTypeConvert(".html", "application/vnd.google-apps.site", "text/html"));

return list;

}

}

}

**CloudUpload Class Code**

using Google.Apis.Auth.OAuth2;

using Google.Apis.Drive.v3;

using Google.Apis.Drive.v3.Data;

using Google.Apis.Services;

using Google.Apis.Util.Store;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace Insert

{

class CloudUpload

{

private UserCredential credential;

private DriveService driveService;

public CloudUpload(string jsonSecretPath, string appName)

{

GetCredential(jsonSecretPath);

CreateDriveService(appName);

}

private void GetCredential(string clientSecretPath)

{

using (var filestream = new FileStream(clientSecretPath,

FileMode.Open, FileAccess.Read))

{

credential = GoogleWebAuthorizationBroker.AuthorizeAsync(

GoogleClientSecrets.Load(filestream).Secrets,

new[] { DriveService.Scope.Drive },

"user",

CancellationToken.None,

new FileDataStore("DriveCommandLineSample")).Result;

}

}

private void CreateDriveService(string applicationName)

{

driveService = new DriveService(new BaseClientService.Initializer()

{

HttpClientInitializer = credential,

ApplicationName = applicationName,

});

}

private void uploadDocumentToDrive(ref byte[] file, string title, string fileId, string description, string mimetype)

{

Google.Apis.Drive.v3.Data.File body = new Google.Apis.Drive.v3.Data.File();

body.Name = title;

body.Description = description;

body.MimeType = mimetype;

fileId = "1rRvc7aKnzKVvvTBwtOnT1-1GO2kz6iK9";

using (System.IO.MemoryStream stream = new System.IO.MemoryStream(file))

{

if (driveService != null)

{

FilesResource.UpdateMediaUpload request = driveService.Files.Update(body, fileId, stream, mimetype);

request.Upload();

Google.Apis.Drive.v3.Data.File uploadedFile = request.ResponseBody;

System.Diagnostics.Debug.WriteLine("Uploaded file: {0} with ID: {1}",

uploadedFile.Name,

uploadedFile.Id);

}

}

}

public bool UploadFile(ref byte[] file, string title, string fileId, string description, string mimetype)

{

bool uploaded = false;

if (credential != null)

{

uploadDocumentToDrive(ref file, title, fileId, description, mimetype);

uploaded = true;

}

return uploaded;

}

}

}

**CloudDownload Class Code**

using Google.Apis.Auth.OAuth2;

using Google.Apis.Download;

using Google.Apis.Drive.v3;

using Google.Apis.Drive.v3.Data;

using Google.Apis.Services;

using Google.Apis.Upload;

using Google.Apis.Util.Store;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Insert

{

class CloudDownload

{

public static UserCredential credential;

public static DriveService driveService;

public CloudDownload(string jsonSecretPath, string appName)

{

GetCredential(jsonSecretPath);

CreateDriveService(appName);

}

private void GetCredential(string clientSecretPath)

{

using (var filestream = new FileStream(clientSecretPath,

FileMode.Open, FileAccess.Read))

{

credential = GoogleWebAuthorizationBroker.AuthorizeAsync(

GoogleClientSecrets.Load(filestream).Secrets,

new[] { DriveService.Scope.Drive },

"user",

CancellationToken.None,

new FileDataStore("DriveCommandLineSample")).Result;

}

}

private void CreateDriveService(string applicationName)

{

driveService = new DriveService(new BaseClientService.Initializer()

{

HttpClientInitializer = credential,

ApplicationName = applicationName,

});

}

private static void convertMemoryStreamToFileStream(MemoryStream stream, string savePath)

{

FileStream fileStream;

using (fileStream = new System.IO.FileStream(savePath, FileMode.OpenOrCreate, FileAccess.Write))

{

try

{

// System.IO.File.Create(saveFile)

stream.WriteTo(fileStream);

fileStream.Close();

}

catch (Exception exc)

{

System.Diagnostics.Debug.WriteLine(exc.Message + " Convert Memory stream Error");

}

}

}

public static void downloadFromDrive(string filename, string fileId, string savePath, string mimeType, Dashboard parentForm)

{

Dashboard parent = parentForm;

try

{

if (Path.HasExtension(filename))

{

var request = driveService.Files.Get(fileId);

var stream = new System.IO.MemoryStream();

System.Diagnostics.Debug.WriteLine(fileId);

request.MediaDownloader.ProgressChanged +=

(IDownloadProgress progress) =>

{

switch (progress.Status)

{

case DownloadStatus.Downloading:

{

System.Diagnostics.Debug.WriteLine(progress.BytesDownloaded);

break;

}

case DownloadStatus.Completed:

{

System.Diagnostics.Debug.WriteLine("Download complete.");

break;

}

case DownloadStatus.Failed:

{

System.Diagnostics.Debug.WriteLine("Download failed.");

MessageBox.Show("File failed to download!!!", "Download Message", MessageBoxButtons.OK, MessageBoxIcon.Error);

break;

}

}

};

request.Download(stream);

convertMemoryStreamToFileStream(stream, savePath + @"\" + @filename);

stream.Dispose();

}

else

{

string extension = "", converter = "";

foreach (MimeTypeConvert obj in MimeConverter.mimeList())

{

if (mimeType == obj.mimeType)

{

extension = obj.extension;

converter = obj.converterType;

}

}

System.Diagnostics.Debug.WriteLine("{0} {1} {2}", fileId, extension, mimeType);

var request = driveService.Files.Export(fileId, converter);

var stream = new System.IO.MemoryStream();

request.MediaDownloader.ProgressChanged +=

(IDownloadProgress progress) =>

{

switch (progress.Status)

{

case DownloadStatus.Downloading:

{

Console.WriteLine(progress.BytesDownloaded);

break;

}

case DownloadStatus.Completed:

{

Console.WriteLine("Download complete.");

break;

}

case DownloadStatus.Failed:

{

Console.WriteLine("Download failed.");

MessageBox.Show("File failed to download!!!", "Download Message", MessageBoxButtons.OK, MessageBoxIcon.Error);

break;

}

}

};

request.Download(stream);

convertMemoryStreamToFileStream(stream, savePath + @"\" + @filename + extension);

stream.Dispose();

}

}

catch (Exception exc)

{

System.Diagnostics.Debug.WriteLine(exc.Message + " Download From Drive Error");

}

}

}

}

**Sync to Google Drive**

private void SynctoCloud\_Click(object sender, EventArgs e)

{

string pathfile = Path.GetDirectoryName(Application.ExecutablePath) + @"\Contacts.accdb";

string filename = "Contacts.accdb";

CloudUpload driveUploader = new CloudUpload(json\_secret\_file, application\_name);

System.Diagnostics.Debug.WriteLine(pathfile);

try

{

byte[] byteArray = System.IO.File.ReadAllBytes(pathfile);

string filePath = pathfile;

string fileName = filename;

string fileId = "1rRvc7aKnzKVvvTBwtOnT1-1GO2kz6iK9";

string description = "test";

string fileType = GetFileType(filePath);

driveUploader.UploadFile(ref byteArray, fileName, fileId, description, fileType);

MessageBox.Show("Sync Successful");

}

catch (Exception exc)

{

System.Diagnostics.Debug.WriteLine(exc.Message);

}

}

### Insert into Access Database

Here is the source code for inserting data into the Microsoft Access database which is used throughout the system. The code is used for all the save buttons in the system. Below is an example of inserting data from the contacts form.

OleDbCommand command = new OleDbCommand("INSERT INTO Contacts (Company,LastName , FirstName, Email, JobTitle, BusinessPhone, HomePhone, MobilePhone, FaxNumber, Address, City, State, Postcode, Country, Webpage, Notes) VALUES(?, ?, ? ,?, ?, ?, ? ,?, ?, ?, ? ,?, ?, ?, ? ,?)", cnon);

command.Parameters.AddWithValue("@Company", Companytxt.Text);

command.Parameters.AddWithValue("@LastName", LastNametxt.Text);

command.Parameters.AddWithValue("@FirstName", FirstNametxt.Text);

command.Parameters.AddWithValue("@Email", Emailtxt.Text);

command.Parameters.AddWithValue("@JobTitle", Jobtxt.Text);

command.Parameters.AddWithValue("@BusinessPhone", BPhonetxt.Text);

command.Parameters.AddWithValue("@HomePhone", Hphonetxt.Text);

command.Parameters.AddWithValue("@MobilePhone", Mphonetxt.Text);

command.Parameters.AddWithValue("@FaxNumber", Faxnotxt.Text);

command.Parameters.AddWithValue("@Address", Addresstxt.Text);

command.Parameters.AddWithValue("@City", Citytxt.Text);

command.Parameters.AddWithValue("@State", Statetxt.Text);

command.Parameters.AddWithValue("@Postcode", Postcodetxt.Text);

command.Parameters.AddWithValue("@Country", Countrytxt.Text);

command.Parameters.AddWithValue("@Webpage", webpagetxt.Text);

command.Parameters.AddWithValue("@Notes", Notestxt.Text);

cnon.Open();

command.Connection = cnon;

command.ExecuteNonQuery();

MessageBox.Show("Save Successful");

cnon.Close();

SaveBtn.Enabled = false;

//

//

//

//

//Adds the new record into the datagridview by loading the database again.

//

string query = "SELECT \* From Contacts";

using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, cnon))

{

DataSet ds = new DataSet();

adapter.Fill(ds);

ContactGridList.DataSource = ds.Tables[0];

}

//

//

//

}

### Read Data from Access Database

This is an example of the source code for reading data from the Microsoft Access database and showing the data in a data grid view. Below is an example of the Contacts Form, reading the ‘Contacts’ table in the database and filling the data grid view with that data on form load.

private void ContactsForm\_Load(object sender, EventArgs e)

{

string query = "SELECT \* From Contacts";

using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, cnon))

{

DataSet ds = new DataSet();

adapter.Fill(ds);

ContactGridList.DataSource = ds.Tables[0];

}

}

### Update Data to Access Database

Updating data from a selected row and inserting it back to the database. Below is the source code of the update button which is used throughout the system and is similar to insert to Access database, with the only difference is being the query.

try

{

String SQLstring = "UPDATE Contacts Set Company = ?, LastName = ?, FirstName = ?, Email = ?, JobTitle = ?, BusinessPhone = ?, HomePhone = ?, MobilePhone = ?, FaxNumber = ?, Address = ?, City = ?, State = ?, Postcode = ?, Country = ?, Webpage = ?, Notes =? WHERE ID = " + IDtxt.Text + "";

using (OleDbCommand command = new OleDbCommand(SQLstring, cnon))

{

command.CommandType = CommandType.Text;

command.Parameters.AddWithValue("@Company", Companytxt.Text);

command.Parameters.AddWithValue("@LastName", LastNametxt.Text);

command.Parameters.AddWithValue("@FirstName", FirstNametxt.Text);

command.Parameters.AddWithValue("@Email", Emailtxt.Text);

command.Parameters.AddWithValue("@JobTitle", Jobtxt.Text);

command.Parameters.AddWithValue("@BusinessPhone", BPhonetxt.Text);

command.Parameters.AddWithValue("@HomePhone", Hphonetxt.Text);

command.Parameters.AddWithValue("@MobilePhone", Mphonetxt.Text);

command.Parameters.AddWithValue("@FaxNumber", Faxnotxt.Text);

command.Parameters.AddWithValue("@Address", Addresstxt.Text);

command.Parameters.AddWithValue("@City", Citytxt.Text);

command.Parameters.AddWithValue("@State", Statetxt.Text);

command.Parameters.AddWithValue("@Postcode", Postcodetxt.Text);

command.Parameters.AddWithValue("@Country", Countrytxt.Text);

command.Parameters.AddWithValue("@Webpage", webpagetxt.Text);

command.Parameters.AddWithValue("@Notes", Notestxt.Text);

cnon.Open();

command.Connection = cnon;

command.ExecuteNonQuery();

MessageBox.Show("Update Successful");

cnon.Close();

}

UpdateBtn.Enabled = false;

SaveBtn.Enabled = false;

AddBtn.Enabled = true;

FirstNametxt.ReadOnly = true;

LastNametxt.ReadOnly = true;

Companytxt.ReadOnly = true;

Emailtxt.ReadOnly = true;

Jobtxt.ReadOnly = true;

BPhonetxt.ReadOnly = true;

Hphonetxt.ReadOnly = true;

Mphonetxt.ReadOnly = true;

Faxnotxt.ReadOnly = true;

Addresstxt.ReadOnly = true;

Citytxt.ReadOnly = true;

Statetxt.ReadOnly = true;

Postcodetxt.ReadOnly = true;

Countrytxt.ReadOnly = true;

webpagetxt.ReadOnly = true;

Notestxt.ReadOnly = true;

string query = "SELECT \* From Contacts";

using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, cnon))

{

DataSet ds = new DataSet();

adapter.Fill(ds);

ContactGridList.DataSource = ds.Tables[0];

}

}

catch (Exception exc)

{

System.Diagnostics.Debug.WriteLine(exc.Message);

}

}

### Delete Data from Access Database

string query = "DELETE from Contacts WHERE ID="+ IDtxt.Text;

using (OleDbCommand command = new OleDbCommand(query, cnon))

{

command.CommandType = CommandType.Text;

cnon.Open();

command.Connection = cnon;

command.ExecuteNonQuery();

MessageBox.Show("Delete Successful");

cnon.Close();

}

string query2 = "SELECT \* From Contacts";

using (OleDbDataAdapter adapter = new OleDbDataAdapter(query2, cnon))

{

DataSet ds = new DataSet();

adapter.Fill(ds);

ContactGridList.DataSource = ds.Tables[0];

}

}

### Search Data grid view feature

Below is the code used in the Contacts Form which searches the First name column based on what the user inputs into the search text. It automatically searches when the user inputs a character and only shows rows similar to what has been inputted based on FirstName.

private void SearchTxt\_TextChanged(object sender, EventArgs e)

{

cnon.Open();

string query = "Select \* from Contacts where FirstName like '" + SearchTxt.Text + "%'";

using (OleDbDataAdapter adapter = new OleDbDataAdapter(query, cnon))

{

DataSet ds = new DataSet();

adapter.Fill(ds);

ContactGridList.DataSource = ds.Tables[0];

cnon.Close();

}

}

### Form Changes

The code shown below is the usual form change code, where the current form hides and the form that has been chosen to be open will load and the old form will close instead of hiding in the background invisible.

private void DashboardBtn\_Click(object sender, EventArgs e)

{

this.Hide();

var dash = new Dashboard();

dash.Closed += (s, args) => this.Close();

dash.Show();

}

private void CalenderBtn\_Click(object sender, EventArgs e)

{

this.Hide();

var cal = new Dashboard();

cal.Closed += (s, args) => this.Close();

cal.Show();

}

private void TaskBtn\_Click(object sender, EventArgs e)

{

this.Hide();

var task = new Dashboard();

task.Closed += (s, args) => this.Close();

task.Show();

}

private void InventoryBtn\_Click(object sender, EventArgs e)

{

this.Hide();

var invent = new Dashboard();

invent.Closed += (s, args) => this.Close();

invent.Show();

}

### Send Email

Sending an email is one of the requirements that would be implemented into the system. Below is the code, which allows the user to send an email.

private void SendBtn\_Click(object sender, EventArgs e)

{

try

{

SmtpClient client = new SmtpClient("smtp.gmail.com", 587);

MailMessage mail = new MailMessage();

mail.From = new MailAddress(Receivertxt.Text);

mail.To.Add(Receivertxt.Text);

mail.Body = Bodytxt.Text;

mail.Subject = SubjectTxt.Text;

if (Attachmenttxt.Text != null)

{

mail.Attachments.Add(new Attachment(Attachmenttxt.Text));

}

client.Credentials = new System.Net.NetworkCredential(emailtxt.Text, passtxt.Text);

client.Send(mail); //Sending Email

MessageBox.Show("Email has been successfully sent.");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, " Message unable to send", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

# Results and Discussion

The system was primarily designed to be a new CRM system that would have more enhanced features than other company CRM’s, so every functionality and feature in the system should be working as intended. Every form had tests done to determine if the final system has met its purpose and requirements, based on every feature added.

## System Testing

A system test will be done to see if the system operates to the desired standard by testing elements on each form. The test table will have columns which are test number, test, event, expected result and actual result as shown in table 10. Within the actual results will reference figures below the test table to prove the actual result. Each test helps with enhancing the system by finding faults within the system. Since each form was created individually and individual features were tested as soon as they were implemented, faults may appear when all of the individual modules are integrated. “This helps to check the combinational behaviour and validate whether the requirements are implemented correctly or not.” (Patel, 2017)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Form being tested** | | | | |
| **Test No.** | **Test** | **Event** | **Expected Result** | **Actual Result** |

Table - Test table headings

### Dashboard

The functionality/features were tested on the dashboard form thoroughly and noted into a test table as shown in table 11. The data that was used was dummy data represented as item 2, item 4 etc. and random numbers as shown in figure 26 and figure 27.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dashboard Form** | | | | |
| **Test No.** | **Test** | **Event** | **Expected Result** | **Actual Result** |
| 1 | Database | System starts up and database file automatically stores into local directory | Database file downloads to local directory of the system. | Same as expected. |
| 2 | Database | Database uploads to google drive | Local Database file uploads to google drive when button pressed and overwrites old file | Same as expected |
| 3 | Contacts button pressed | Form changes to contacts form when contacts button pressed | Contacts Form successfully loads | Same as expected. |
| 4 | Calendar button pressed | Form changes to contacts form when contacts button pressed | Calendar Form successfully loads | Same as expected. |
| 5 | Tasks button pressed | Form changes to contacts form when contacts button pressed | Tasks Form successfully loads | Same as expected. |
| 6 | Inventory Management button pressed | Form changes to contacts form when contacts button pressed | Inventory Management Form successfully loads | Same as expected. |
| 7 | Email button pressed | Email Form Loads | Email Form successfully loads | Same as expected. |
| 8 | Chart | Graphs read from database | Graph updates accordingly to database | Same as expected  See Figure 25 & Figure 26 |
| 9 | Error catch | Input something wrong, message pops up warning user | Message box appears to inform the user, what went wrong and does not crash the system. | Same as expected |

Table - Dashboard Form Test Table

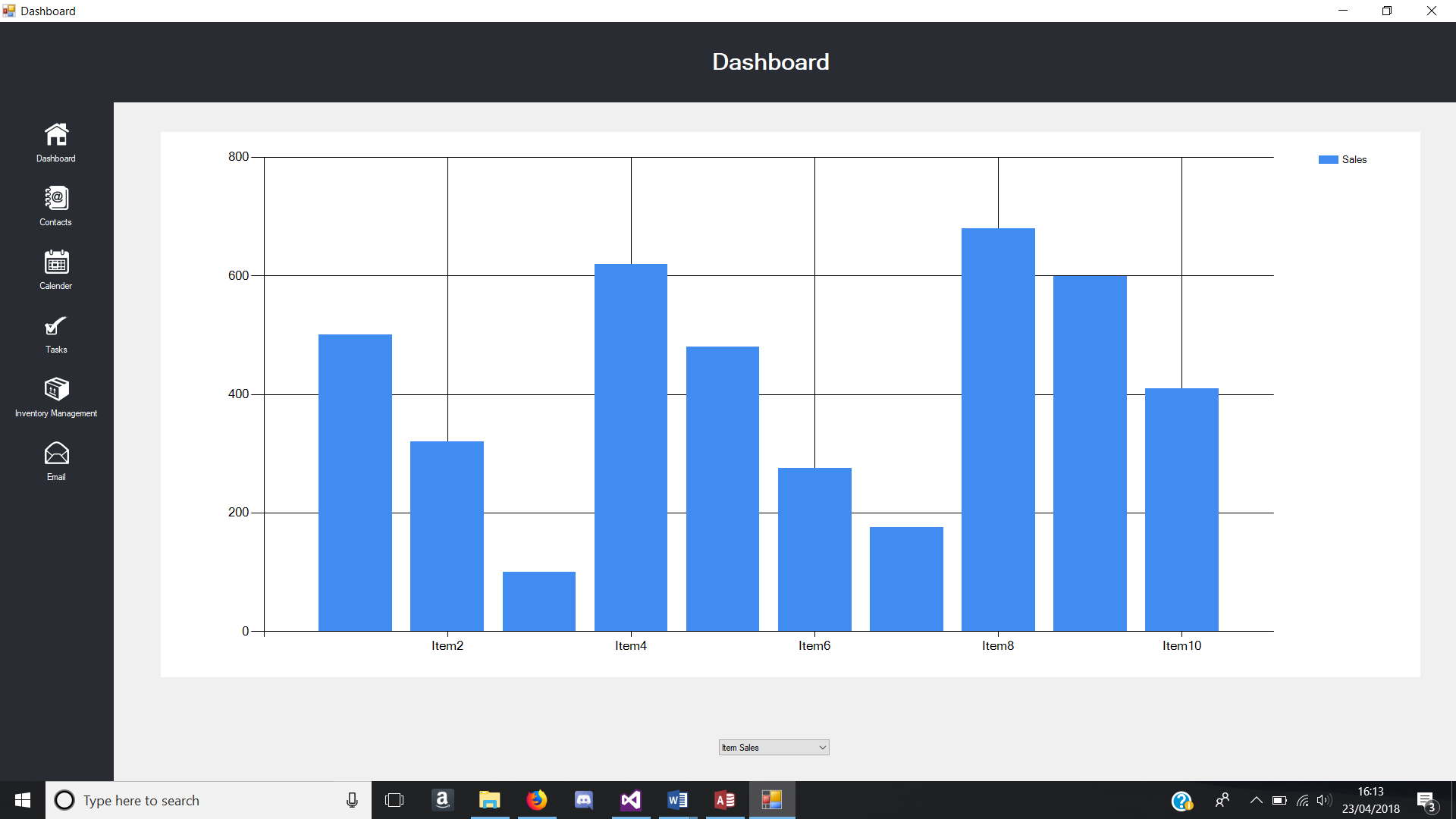


Figure - Chart change test with dummy data

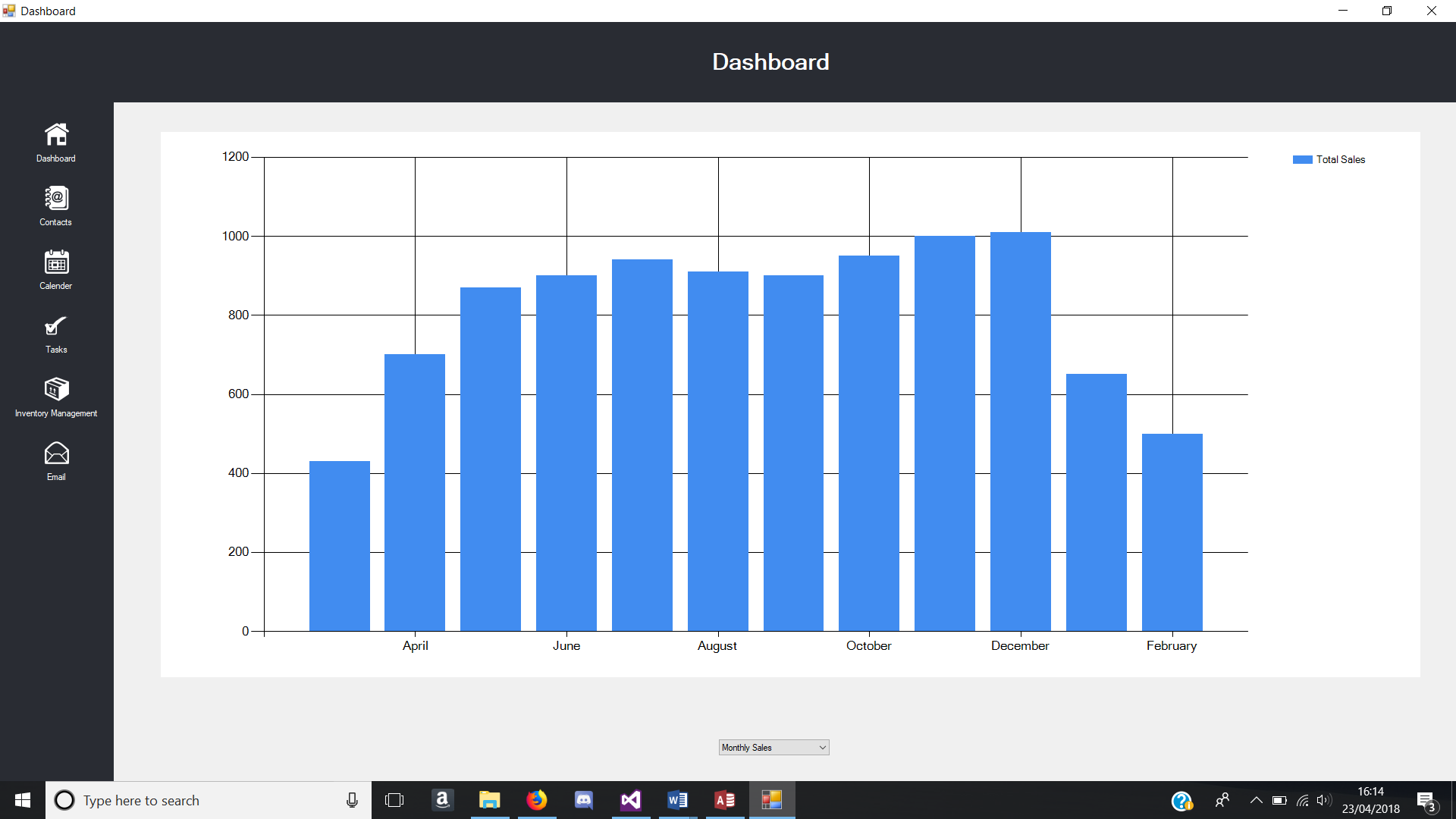


Figure - Chart change test with dummy data

### Contacts Form

The functionality/features were tested on the dashboard form thoroughly and noted into a test table as shown in table 11. The data that was used during this test was dummy data in every column of data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Contacts Form** | | | | |
| **Test No.** | **Test** | **Event** | **Expected Result** | **Actual Result** |
| 1 | Database | Database is read successfully into data grid view | Data grid view is populated with data from database on load | Same as expected. |
| 2 | Database | Database uploads to google drive | Local Database file uploads to google drive when button pressed and overwrites old file | Same as expected |
| 3 | Calendar button pressed | Form changes to contacts form when contacts button pressed | Calendar Form successfully loads | Same as expected. |
| 4 | Tasks button pressed | Form changes to contacts form when contacts button pressed | Tasks Form successfully loads | Same as expected. |
| 5 | Inventory Management button pressed | Form changes to contacts form when contacts button pressed | Inventory Management Form successfully loads | Same as expected. |
| 6 | Dashboard Button press | Form changes to contacts form when contacts button pressed | Dashboard Form successfully loads | Same as expected. |
| 7 | Email button pressed | Email Form Loads | Email Form successfully loads | Same as expected. |
| 8 | Add button pressed | Textboxes are enabled for user to input data when add button is pressed | Textboxes are enabled for user to input data and browse button enabled. | Same as expected.  See Figure 28 |
| 9 | Save button pressed | Data inputted are saved into the database when save button is pressed. | Data inputted in the textboxes are saved into the database. | Same as expected.  See Figure 29 & 30 |
| 10 | Edit button pressed | Textboxes enabled and save and add button are disabled | Textboxes are enabled for user to input data when edit button is pressed. Add and Save button are disabled. | Same as expected  See Figure 3 2& 33 |
| 11 | Update button pressed | Data that has been changed in textboxes are saved into the database. | Changed data inputted in the textboxes are saved into the database. | Same as expected.  See Figure 34 |
| 12 | Delete button pressed | Removes row from database | Row of data from the database is deleted | Same as expected.  See Figure 35 |
| 13 | Data grid view | Data grid view row selected, and textboxes are filled with corresponding data. | Textboxes are filled with data when data grid view row is selected | Same as expected.  See Figure 31 |
| 14 | Validation | Particular textboxes have validations | Textboxes that require integer will only allow integer inputs | Same as expected |
| 15 | Error catch | Input something wrong, message pops up warning user | Message box appears to inform the user, what went wrong and does not crash the system. | Same as expected |
| 16 | Form Maximised | Interface changes appropriately when maximised | All form controls move to the centre when form is maximised | Same as expected |
| 17 | Sync to google drive | Database file is edited in google drive with the latest version | Database file is edited in google drive with the latest version | Same as expected |
| 18 | Search | Filter through data and show what the has been inputted in the textbox | Data is filtered and is displayed in the data grid view | Same as expected  See figure 36 |

Table - Contacts Form Test Table

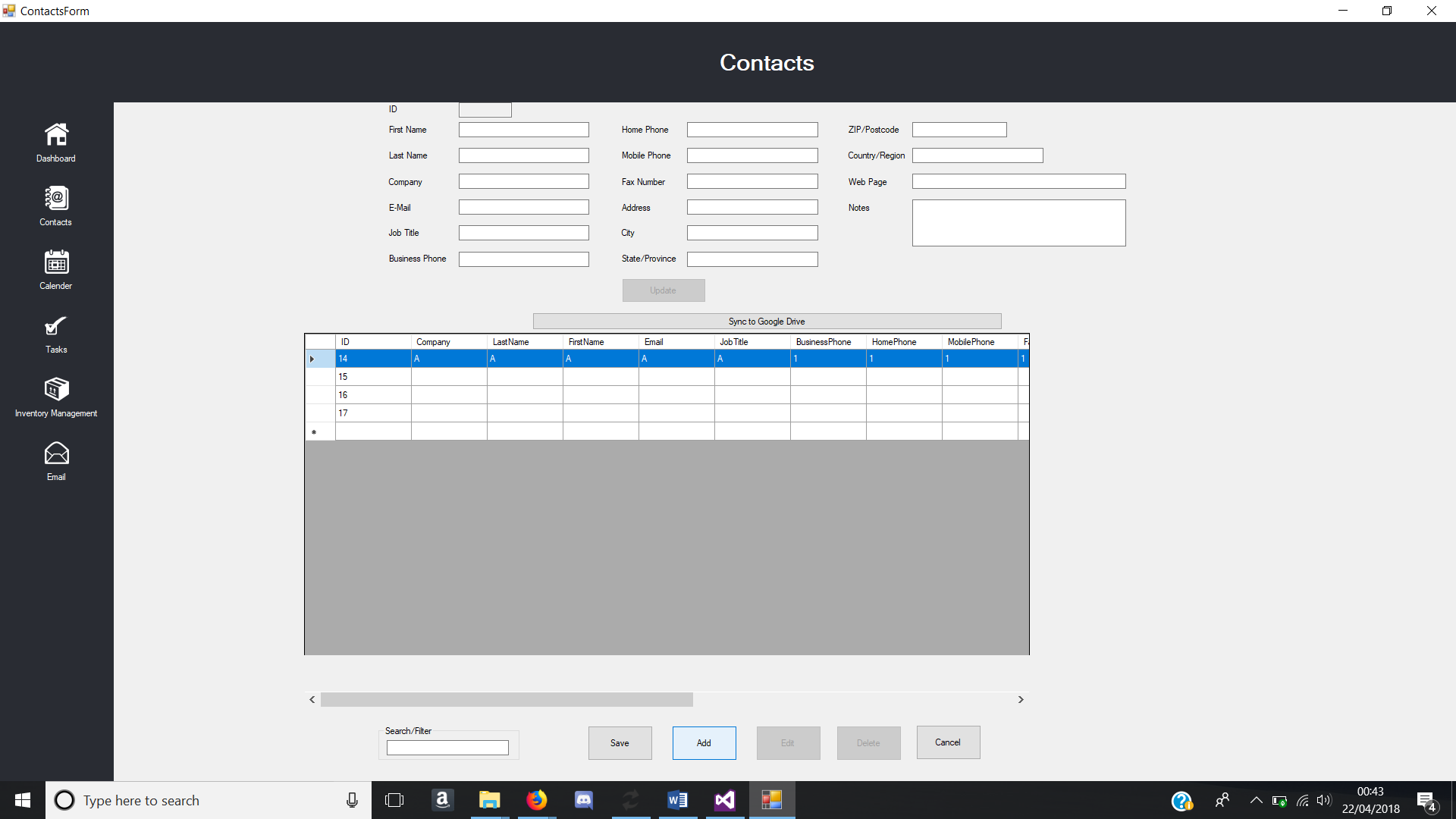


Figure - Add Function Test

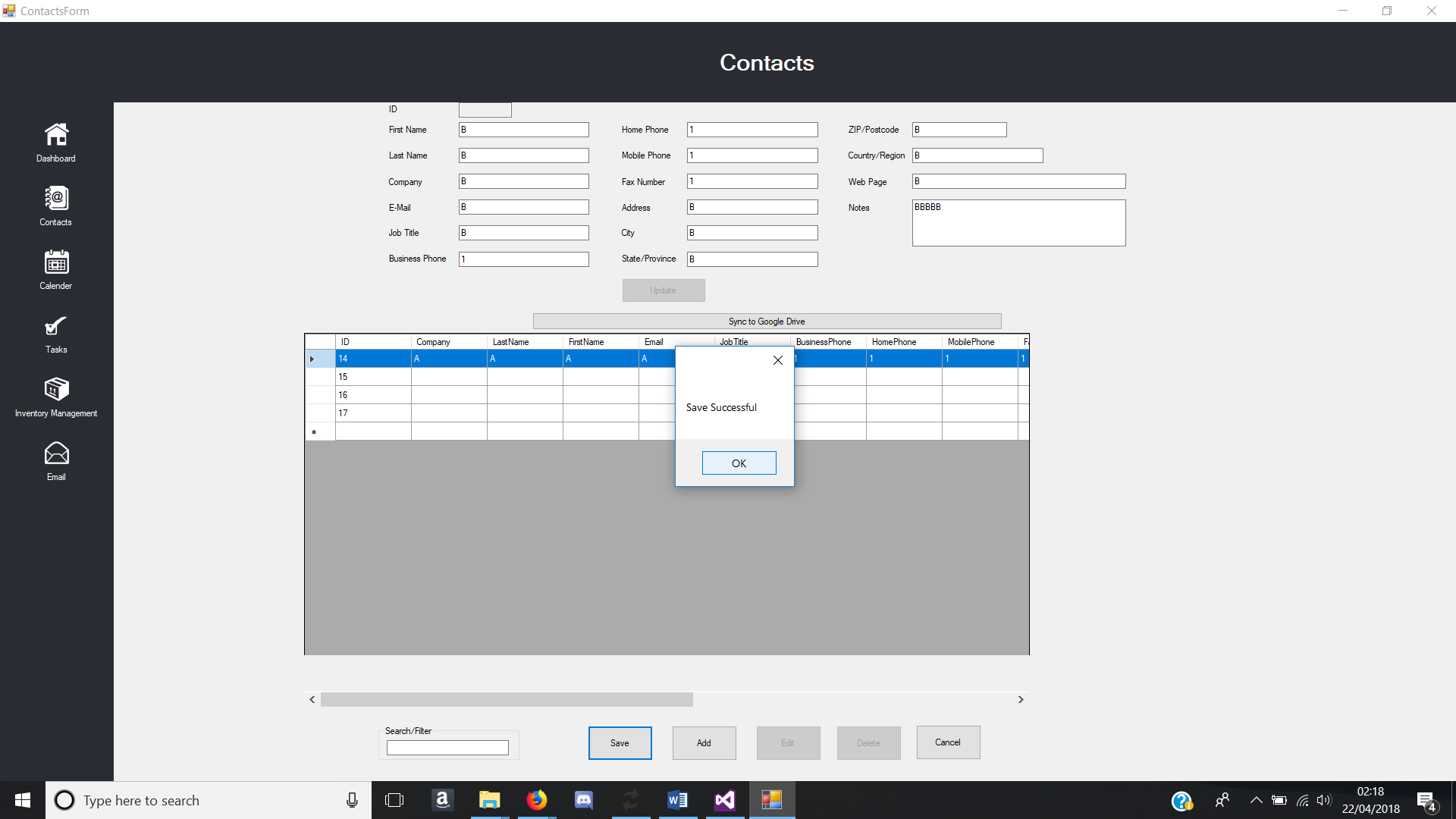


Figure - Save Button pressed when textboxes are filled with data

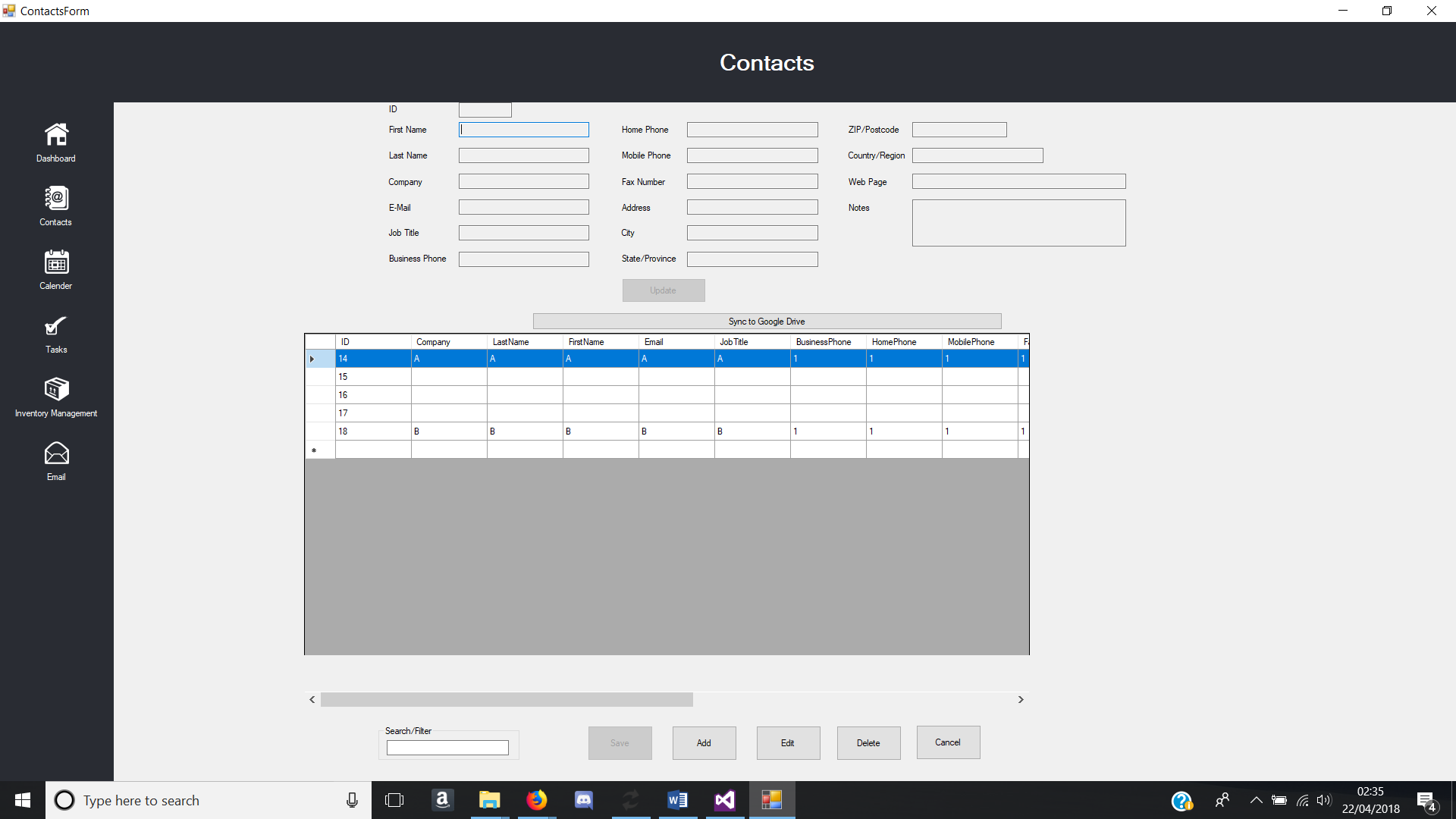


Figure - Data inserted into database and loaded into data grid view



Figure - Selected record data outputted to correct textboxes

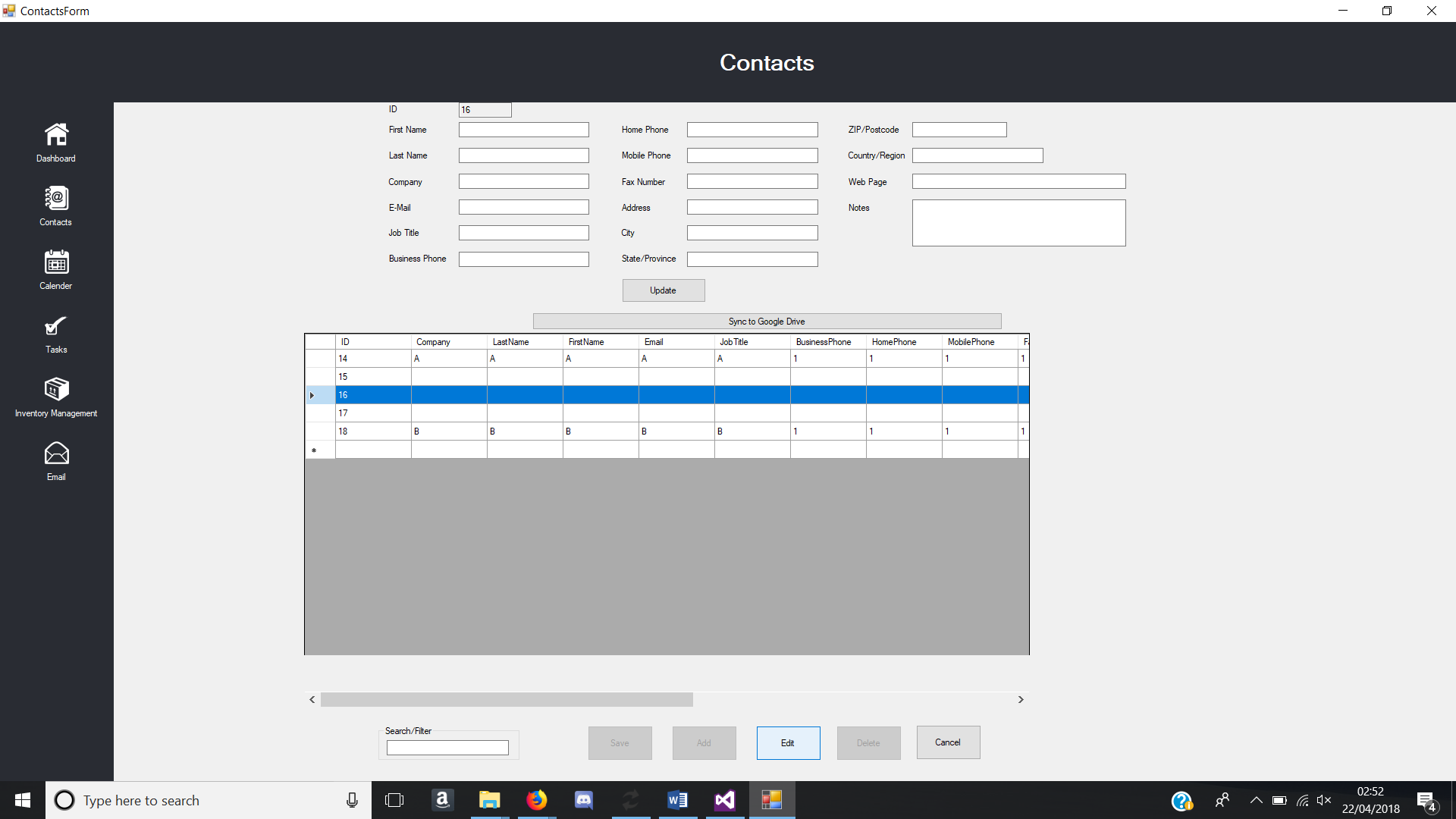


Figure - Edit function test

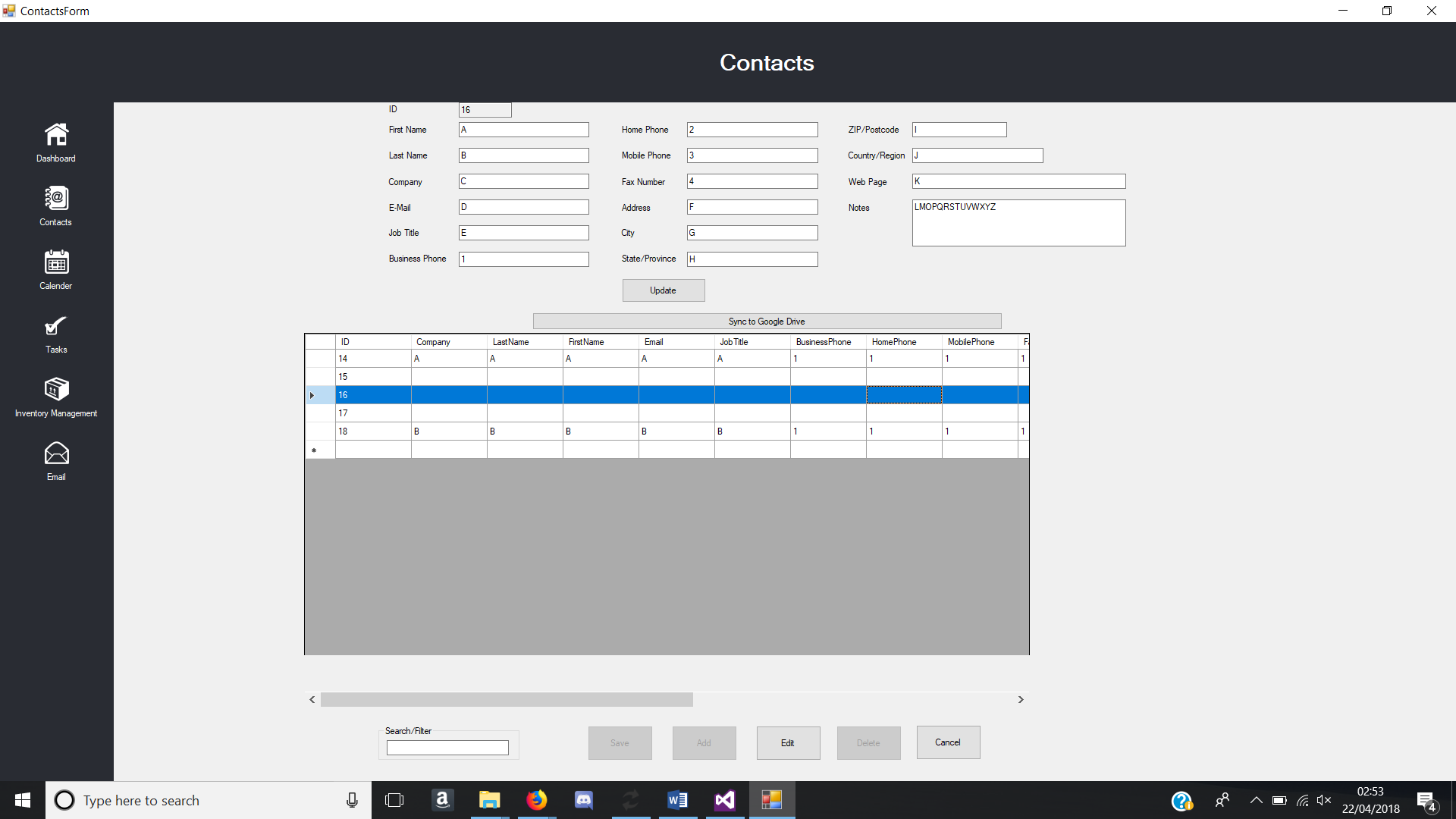


Figure - Data being changed in textbox

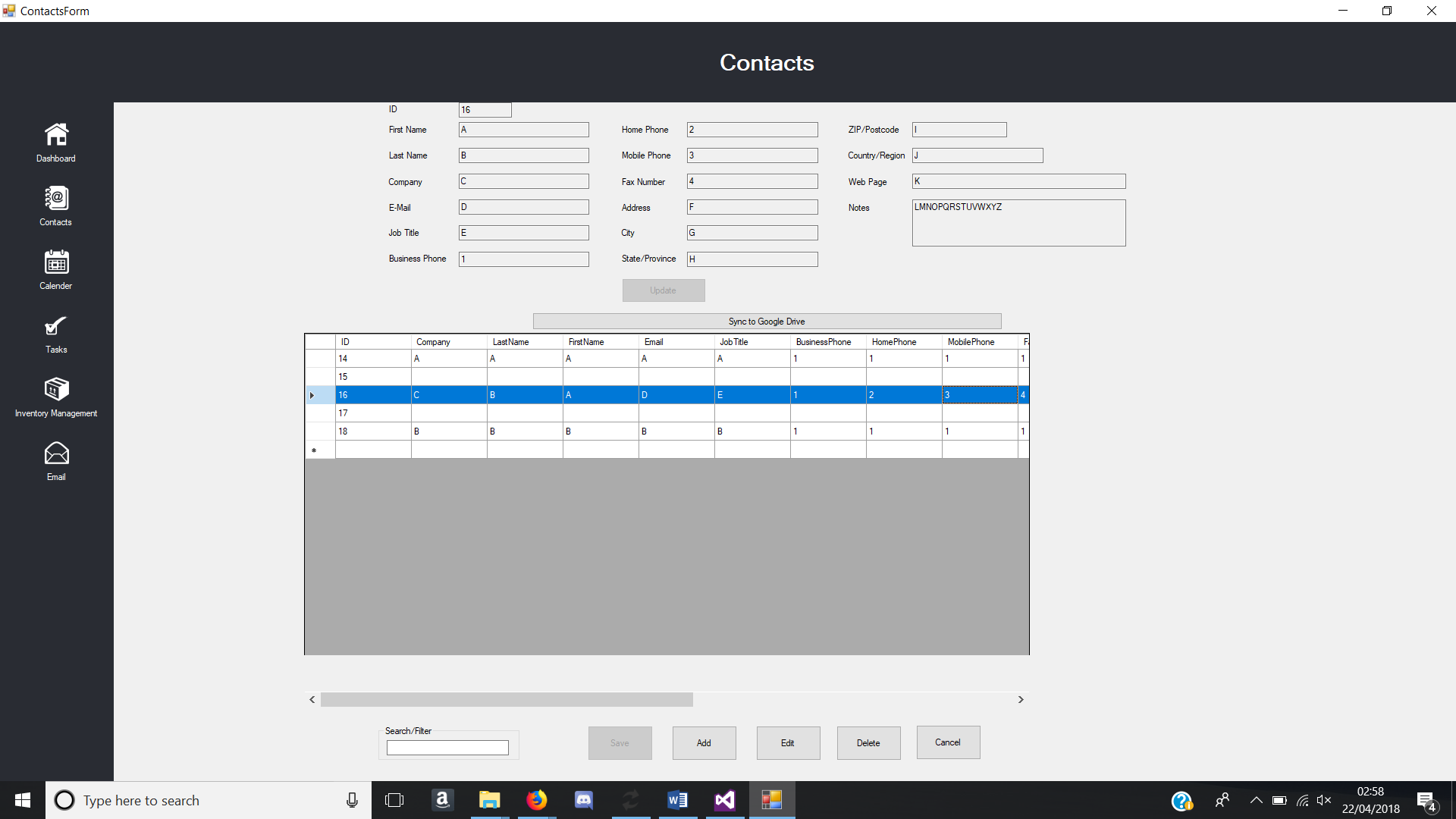


Figure - Update function test

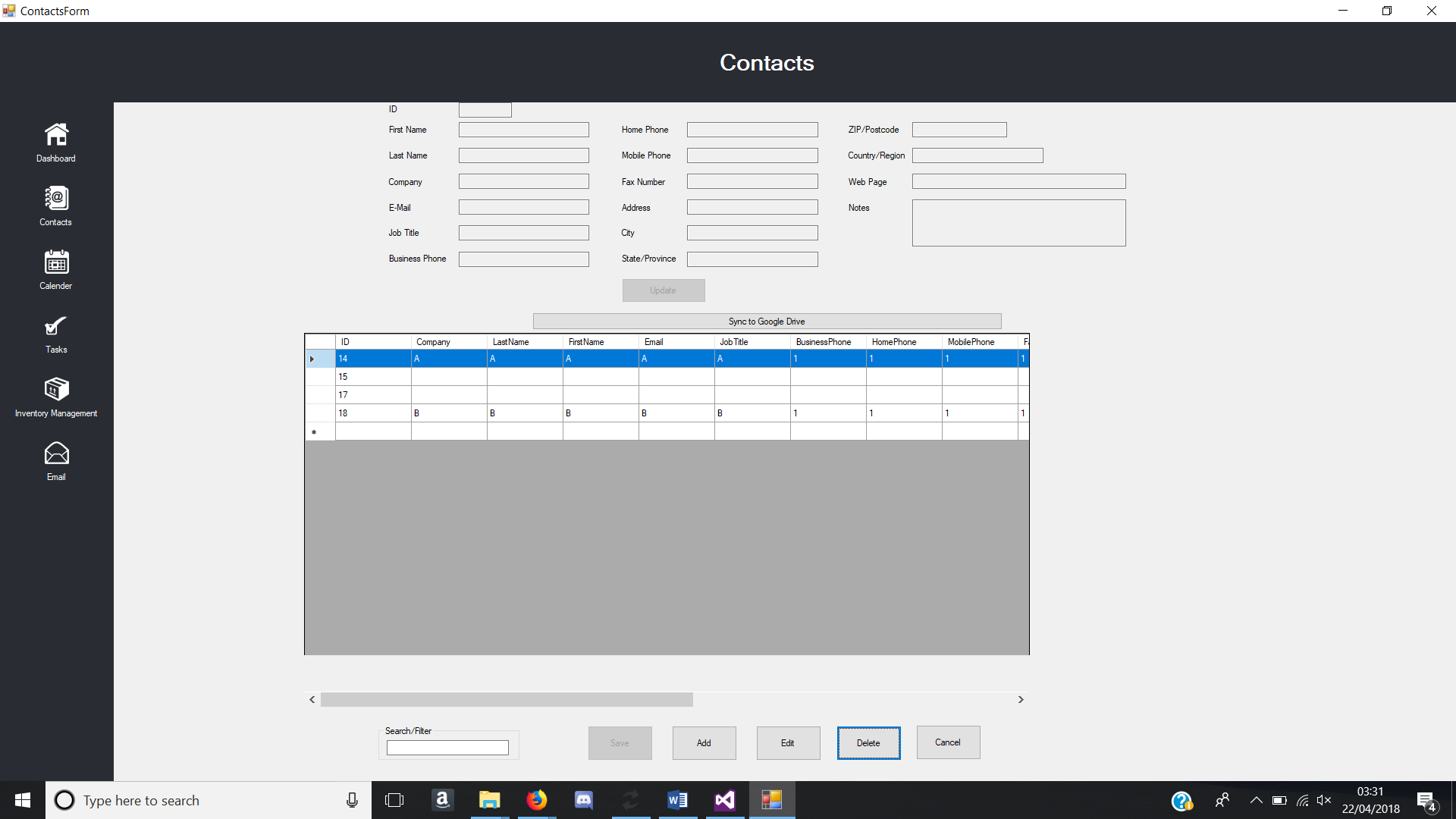


Figure - Delete Function test.

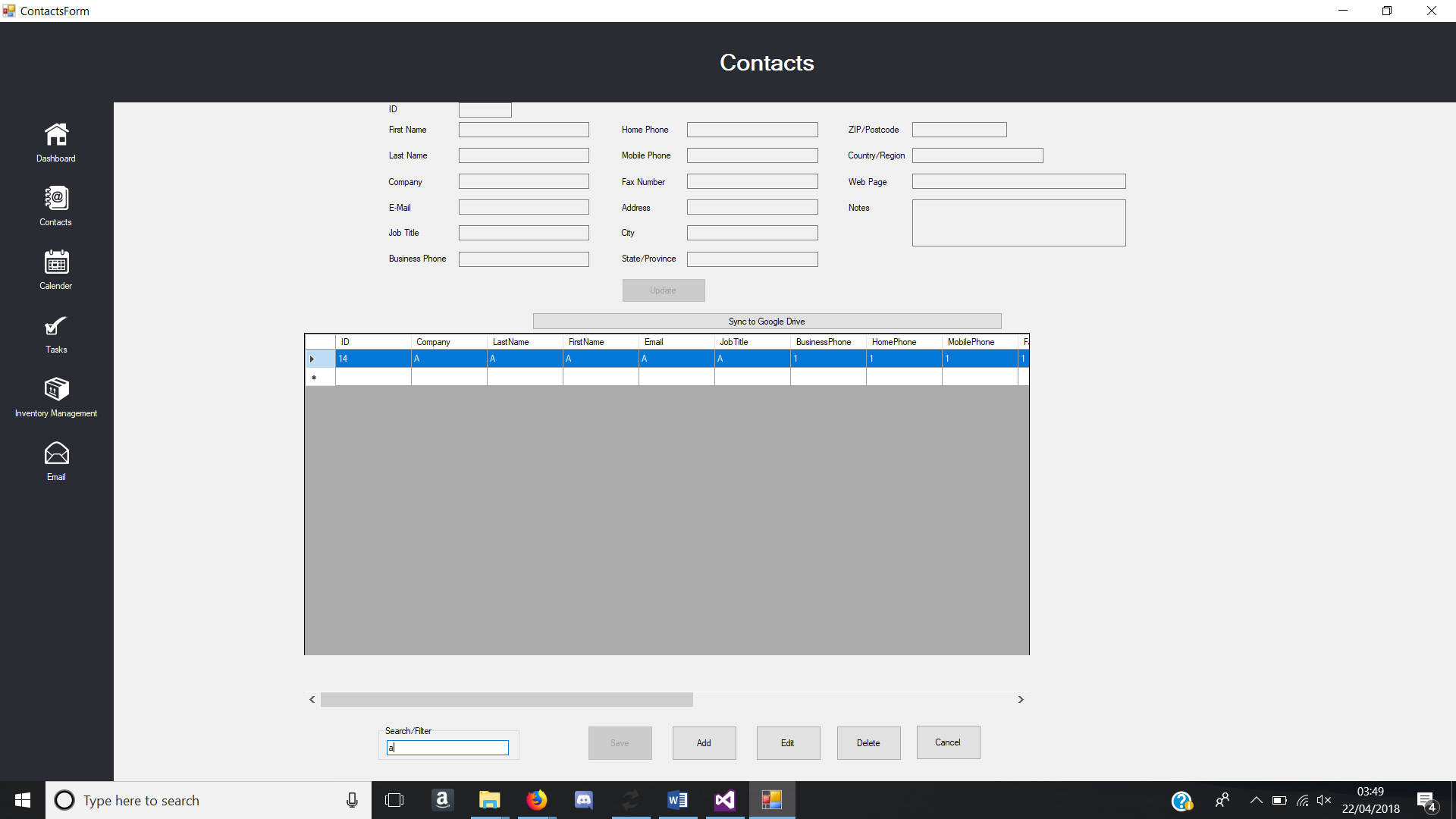


Figure - Search Function test

### Calendar Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Calendar Form** | | | | |
| **Test No.** | **Test** | **Event** | **Expected Result** | **Actual Result** |
| 1 | Database | Database is read successfully into data grid view | Data grid view is populated with data from database on load | Same as expected. |
| 2 | Database | Database uploads to google drive | Local Database file uploads to google drive when button pressed and overwrites old file | Creates a new file in google drive instead of replacing the old file. |
| 3 | Contacts button pressed | Form changes to contacts form when contacts button pressed | Contacts Form successfully loads | Same as expected. |
| 4 | Tasks button pressed | Form changes to contacts form when contacts button pressed | Tasks Form successfully loads | Same as expected. |
| 5 | Inventory Management button pressed | Form changes to contacts form when contacts button pressed | Inventory Management Form successfully loads | Same as expected. |
| 6 | Dashboard Button press | Form changes to contacts form when contacts button pressed | Dashboard Form successfully loads | Same as expected. |
| 7 | Email button pressed | Email Form Loads | Email Form successfully loads | Same as expected. |
| 8 | Add button pressed | Textboxes are enabled for user to input data when add button is pressed | Textboxes are enabled for user to input data and browse button enabled. | Same as expected.  See figure 37 |
| 9 | Save button pressed | Data inputted are saved into the database when save button is pressed. | Data inputted in the textboxes and image chosen are saved into the database. | Same as expected. |
| 10 | Edit button pressed | Textboxes enabled and save and add button are disabled | Textboxes are enabled for user to input data when edit button is pressed. Add and Save button are disabled. | Same as expected |
| 11 | Delete button pressed | Removes row from database | Row of data from the database is deleted | Same as expected. |
| 12 | Data grid view | Data grid view row selected, and textboxes are filled with corresponding data. | Textboxes are filled with data when data grid view row is selected | Same as expected. |
| 13 | Validation | Particular textboxes have validations | Textboxes such as duration textbox allows integer only inputs | Same as expected |
| 14 | Error catch | Input something wrong, message pops up warning user | Message box appears to inform the user, what went wrong and does not crash the system. | Same as expected |
| 15 | Form Maximised | Interface changes appropriately when maximised | All form controls move to the centre when form is maximised | Same as expected |

Table - Calendar Form Test Table

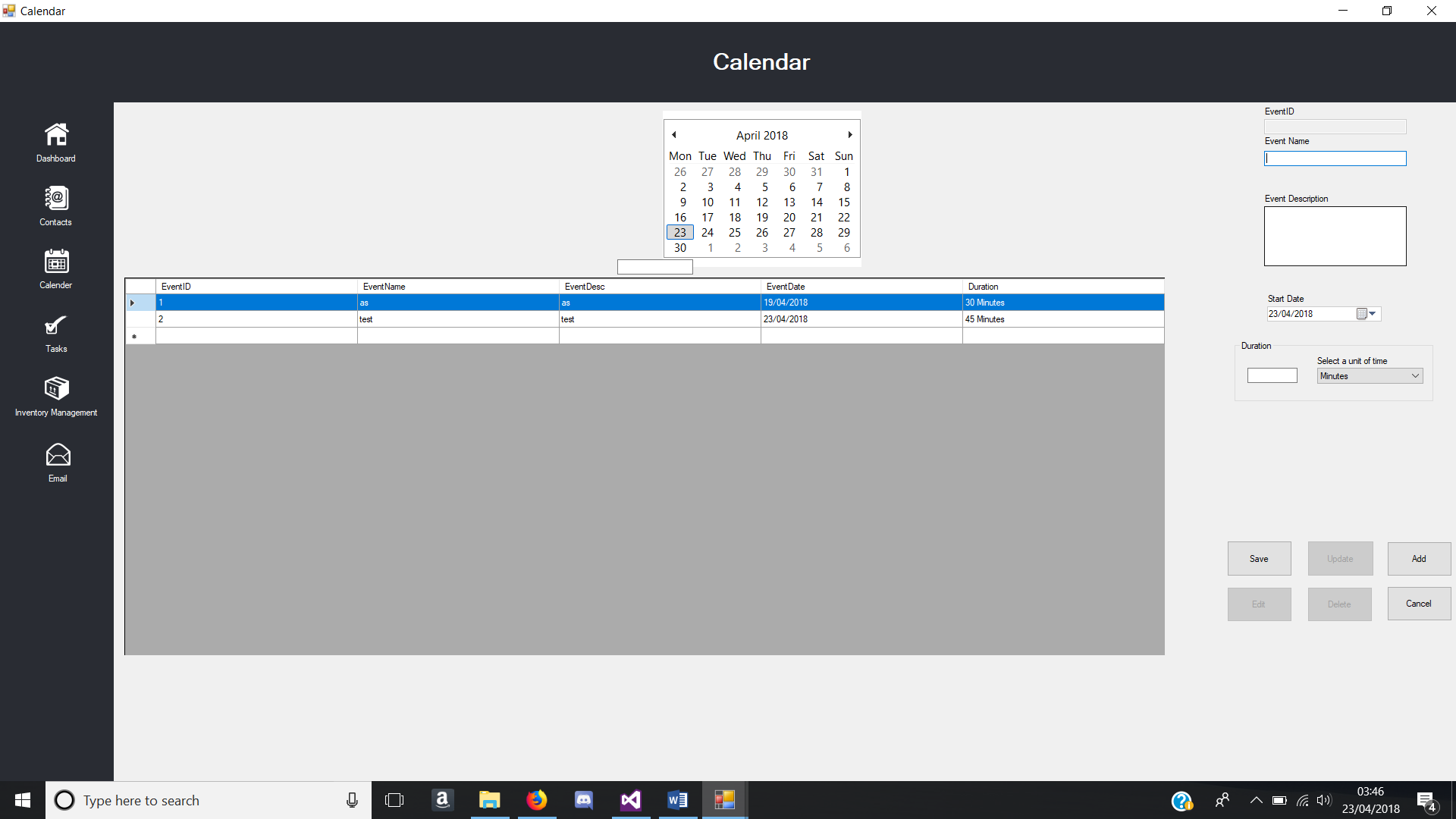


Figure - Add button pressed test

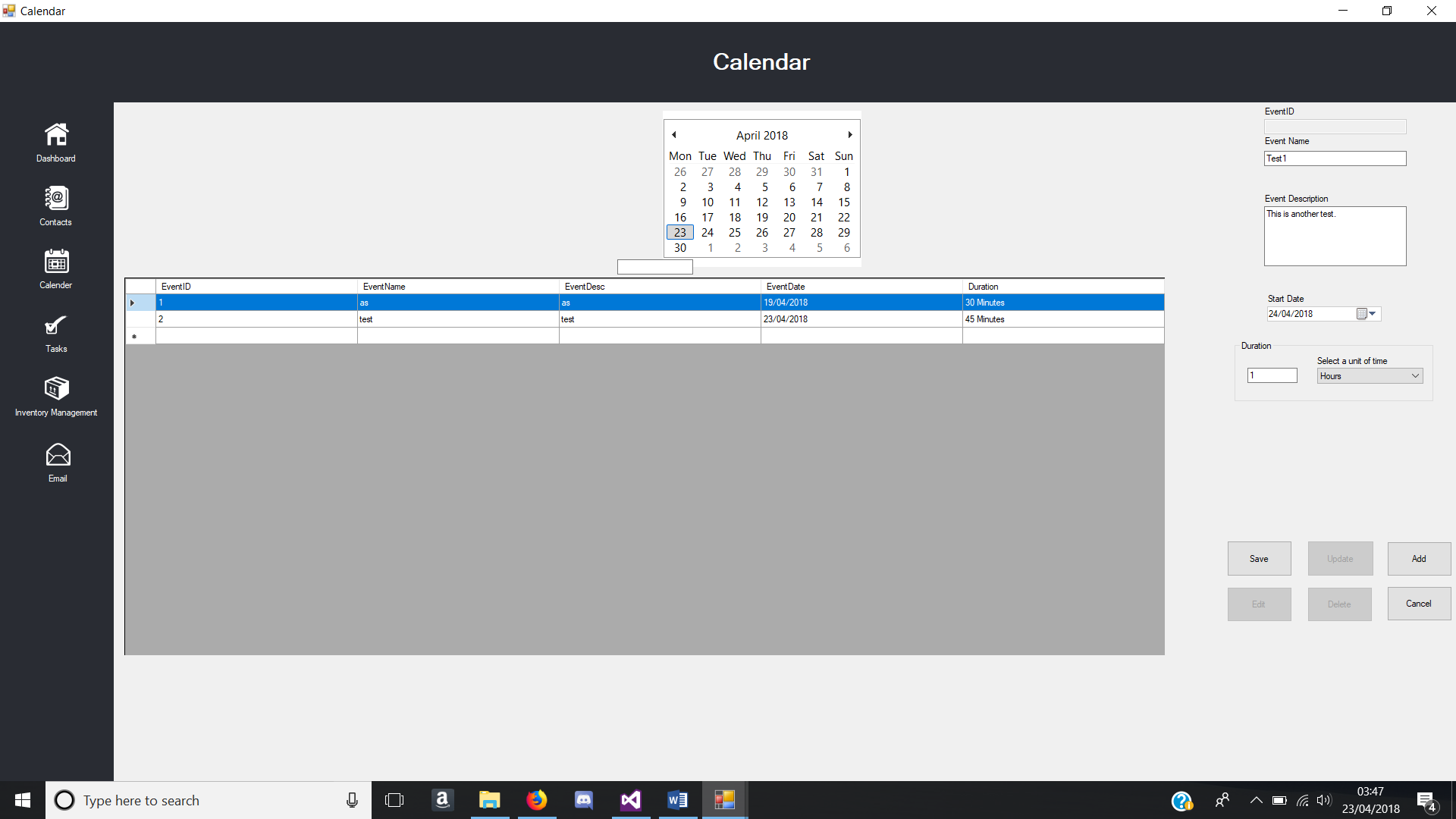


Figure - Data inputted in textboxes and date and dropdown list is selected

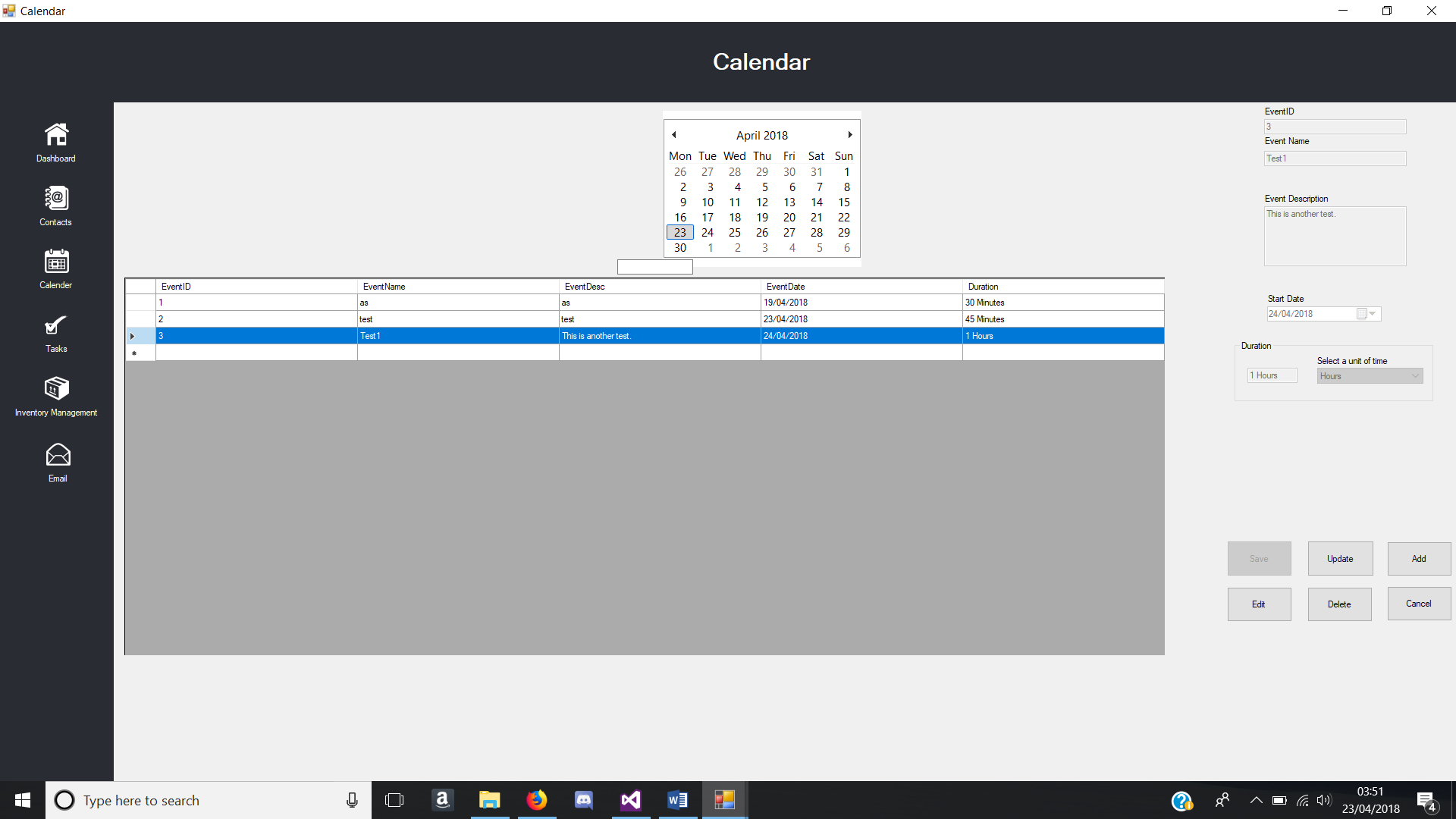


Figure - Save button press Test

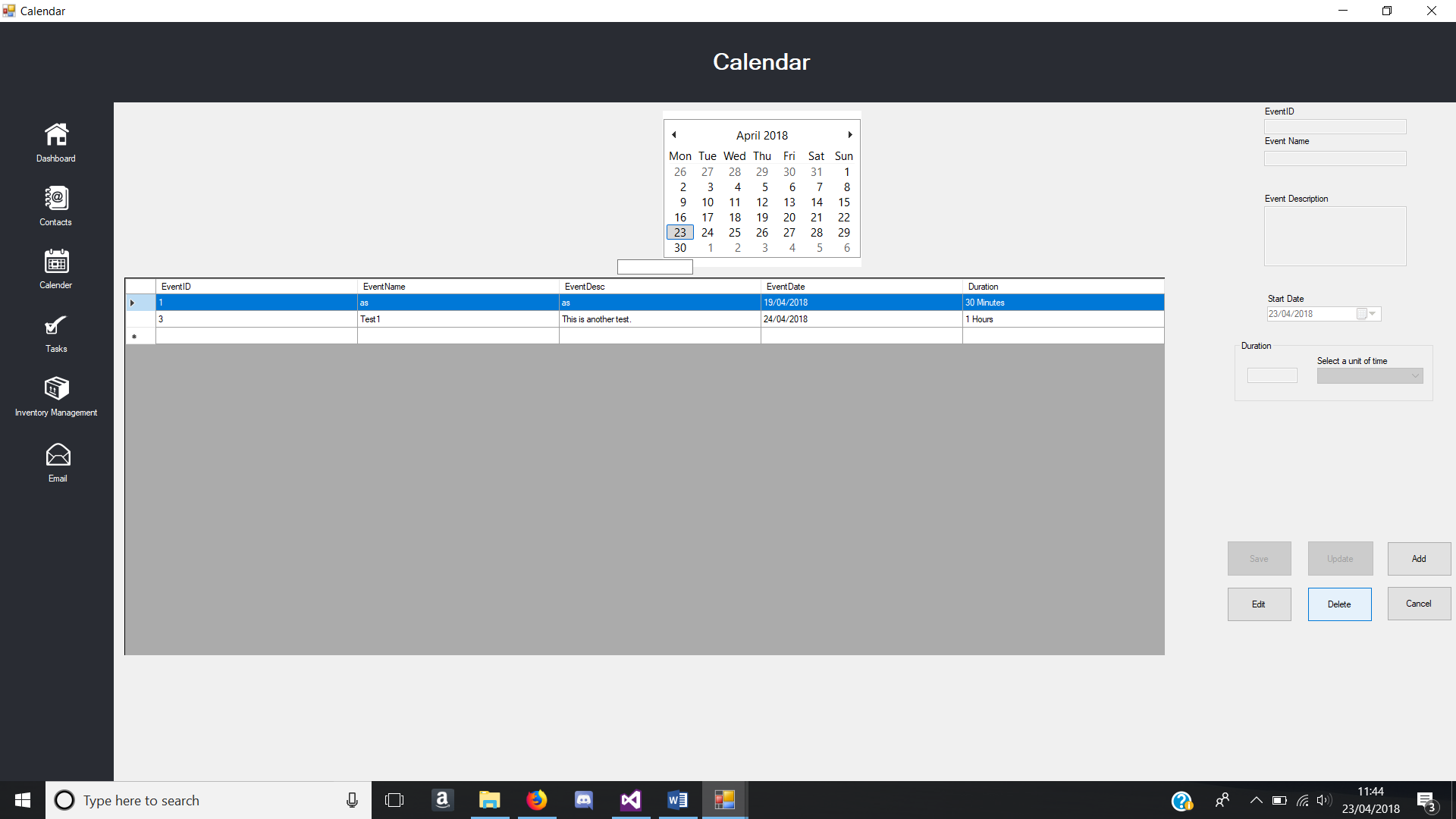


Figure - Delete button pressed test

### Tasks Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tasks Form** | | | | |
| **Test No.** | **Test** | **Event** | **Expected Result** | **Actual Result** |
| 1 | Database | System starts up and database file automatically stores into local directory | Database file downloads to local directory of the system. | Same as expected. |
| 2 | Database | Database uploads to google drive | Local Database file uploads to google drive when button pressed and overwrites old file | Same as expected |
| 3 | Contacts button pressed | Form changes to contacts form when contacts button pressed | Contacts Form successfully loads | Same as expected. |
| 4 | Calendar button pressed | Form changes to contacts form when contacts button pressed | Calendar Form successfully loads | Same as expected. |
| 5 | Inventory Management button pressed | Form changes to contacts form when contacts button pressed | Inventory Management Form successfully loads | Same as expected. |
| 6 | Dashboard Button press | Form changes to contacts form when contacts button pressed | Dashboard Form successfully loads | Same as expected. |
| 7 | Email button pressed | Email Form Loads | Email Form successfully loads | Same as expected. |
| 8 | Add button pressed | Textboxes are enabled for user to input data when add button is pressed | Textboxes are enabled for user to input data and browse button enabled. | Same as expected.  See Figure 41 |
| 9 | Save button pressed | Data inputted are saved into the database when save button is pressed. | Data inputted in the textboxes and selected date in date picker are saved into the database. | Same as expected  See figure 42 & figure 43 |
| 10 | Edit button pressed | Textboxes enabled and save and add button are disabled | Textboxes are enabled for user to input data when edit button is pressed. Add and Save button are disabled. | Same as expected  See figure 44 |
| 11 | Update button pressed | Changed data in textboxes are saved into the same record in the database. | The edited data in textboxes are saved into the record that was selected when edit button was pressed. | Same as expected  See figure 45 |
| 12 | Delete button pressed | Removes row from database | Row of data from the database is deleted | Same as expected.  See figure 46 & figure 47 |
| 13 | Month Calendar Date Selection | Reads from the database where Start date equals the selected date. | Data grid view is loaded with correct data with the selected date. | Same as expected. |
| 14 | Data grid view | Data grid view row selected, and textboxes are filled with corresponding data. | Textboxes are filled with data when data grid view row is selected | Same as expected. |
| 15 | Validation | Particular textboxes have validations | Specific textbox must be filled or message box appears | Same as expected |
| 16 | Error catch | Input something wrong, message pops up warning user | Message box appears to inform the user, what went wrong and does not crash the system. | Same as expected |
| 17 | Form Maximised | Interface changes appropriately when maximised | All form controls move to the centre when form is maximised | Same as expected |

Table - Tasks Form Test Table

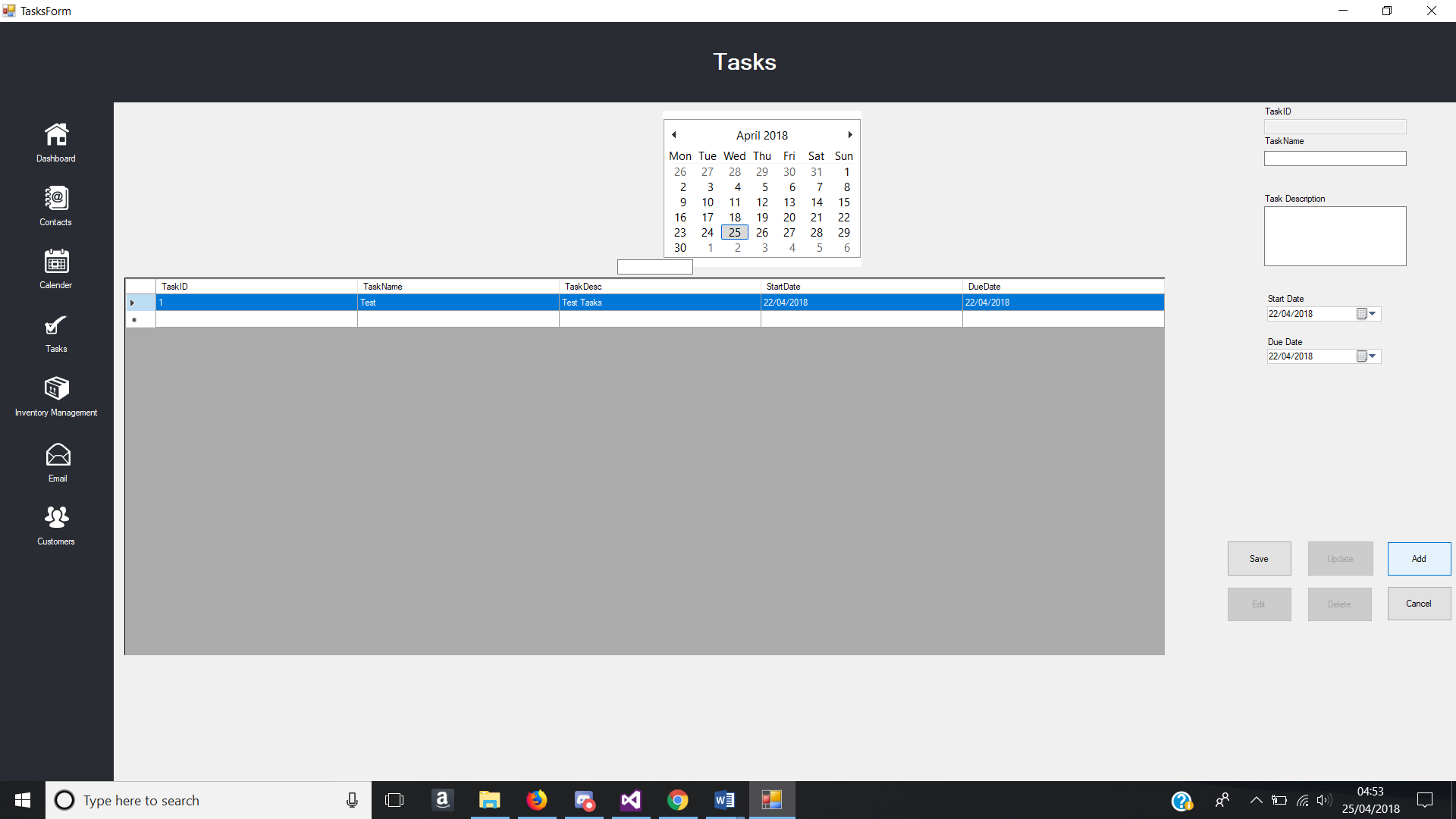


Figure - Add Button Pressed Test

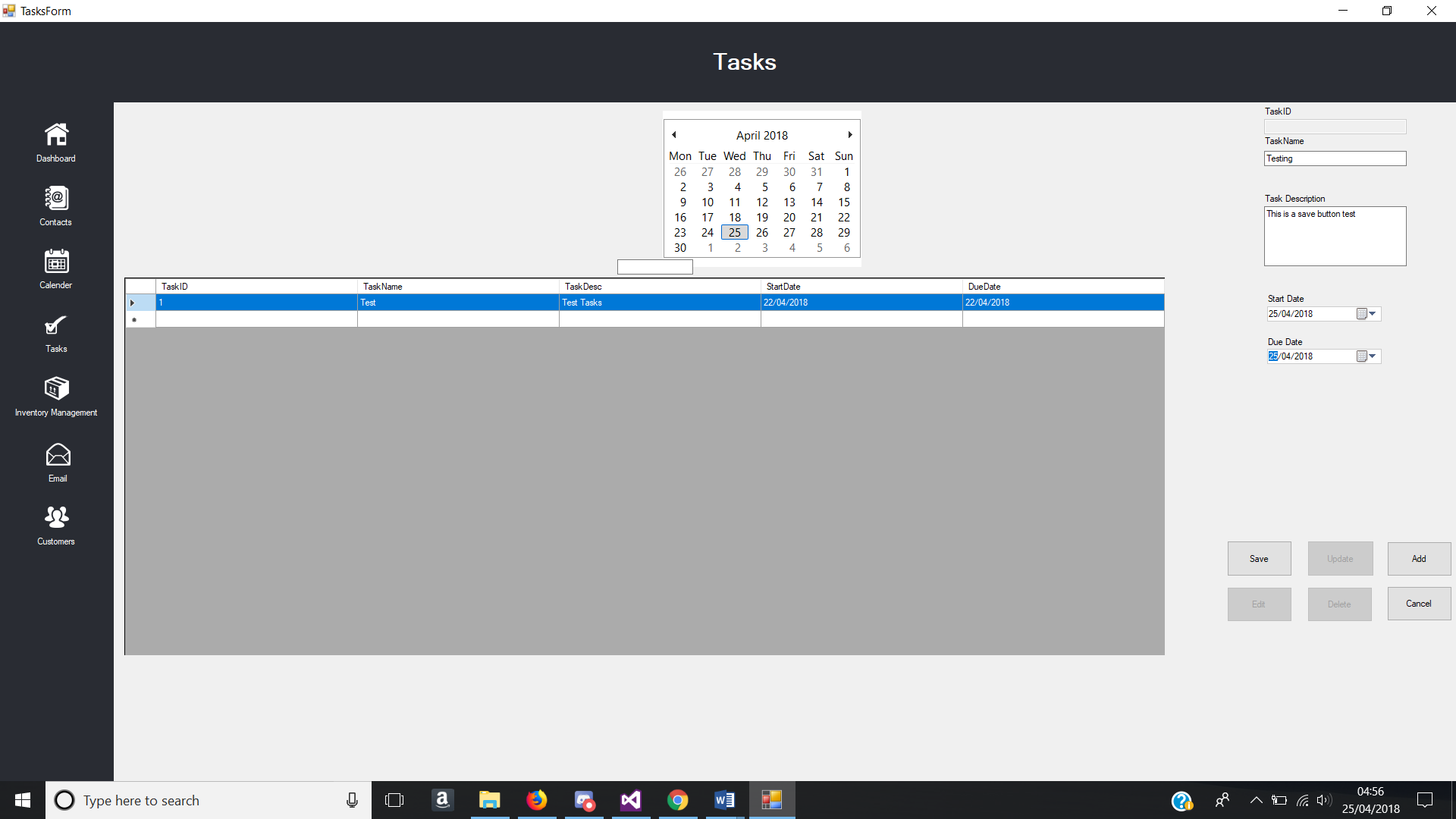


Figure - Data inputted into textbox and Dates selected

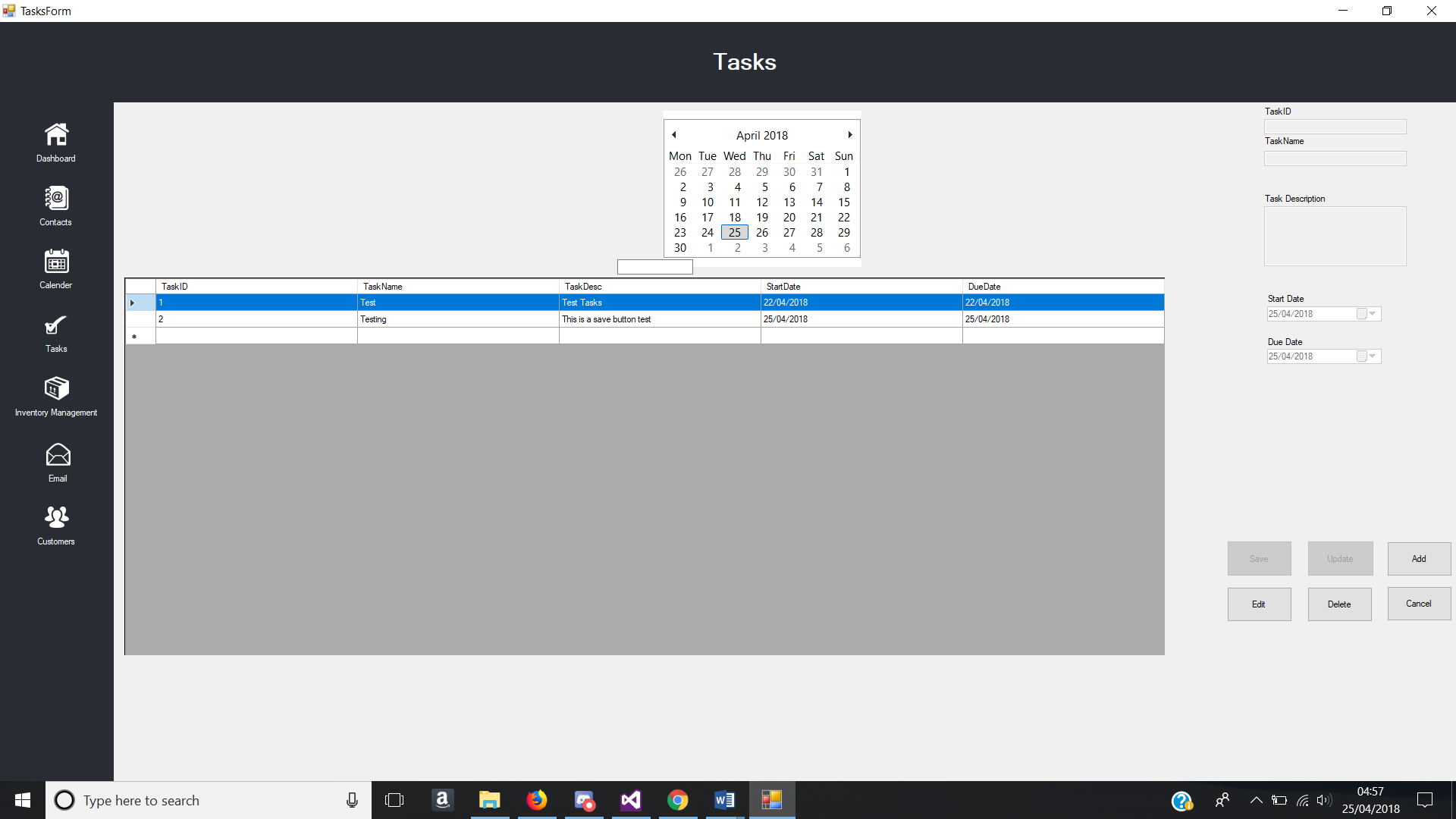


Figure - Save Button Pressed Test

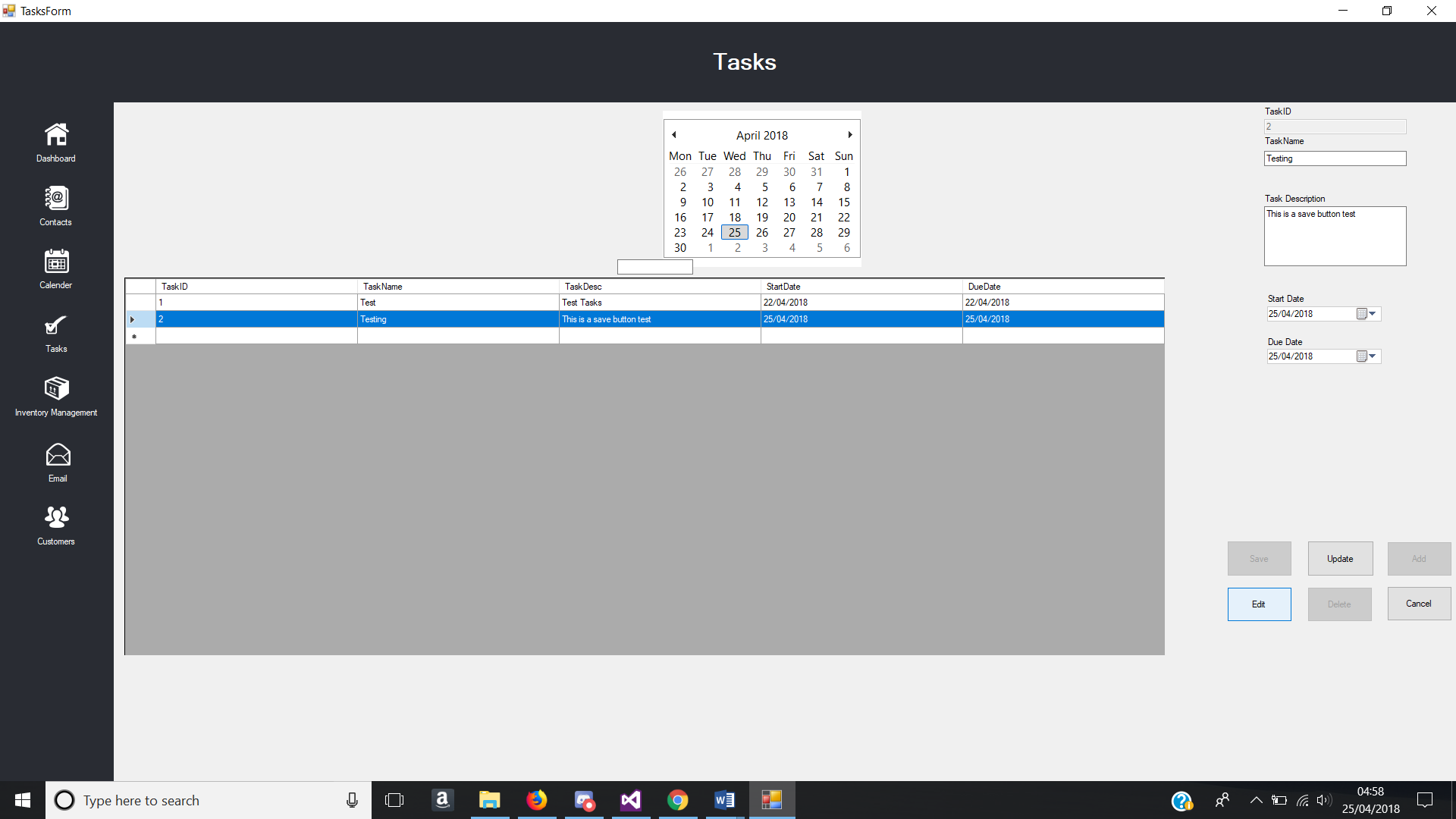


Figure - Edit Button Pressed Test

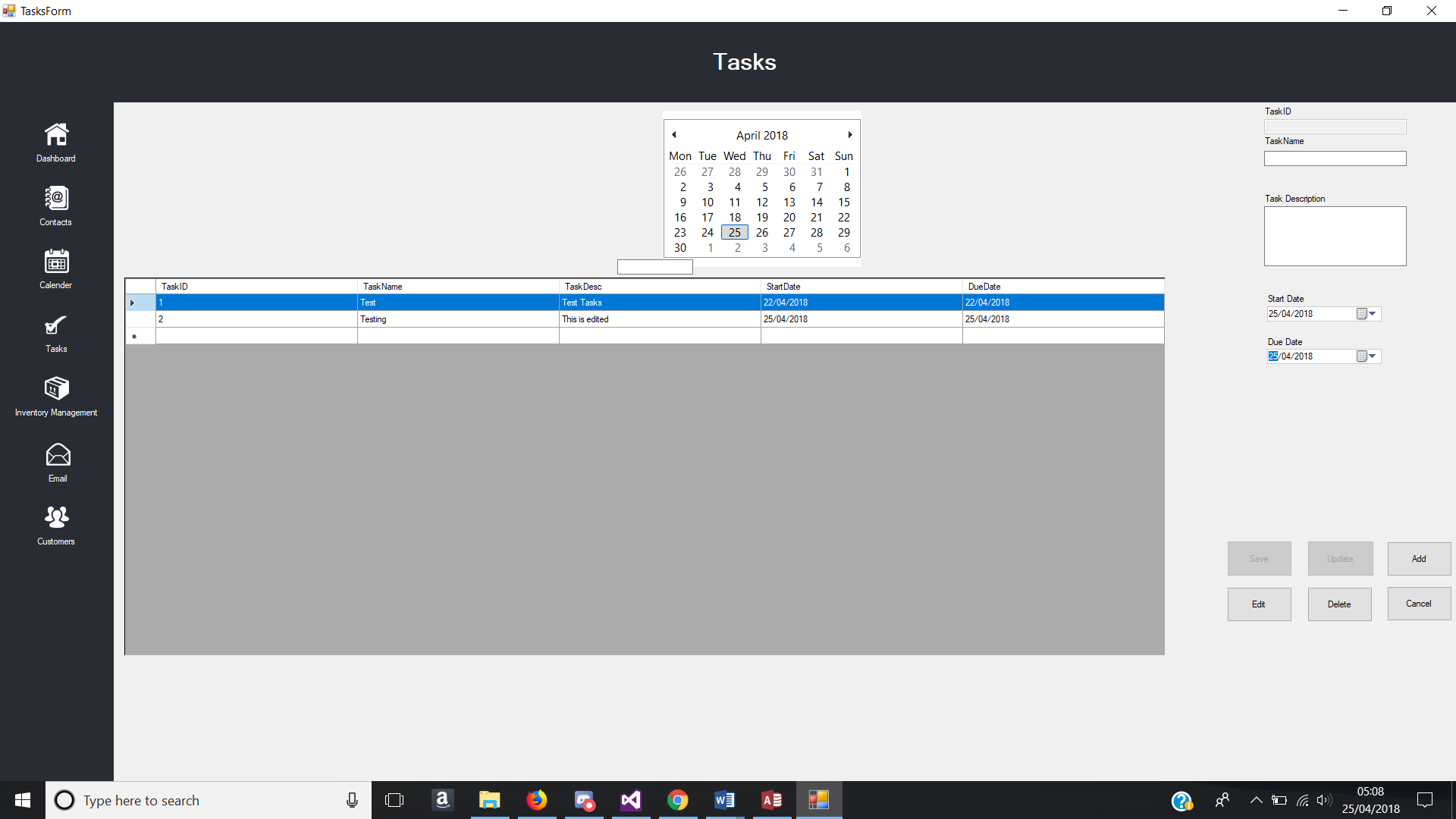


Figure - Update Button Pressed Test

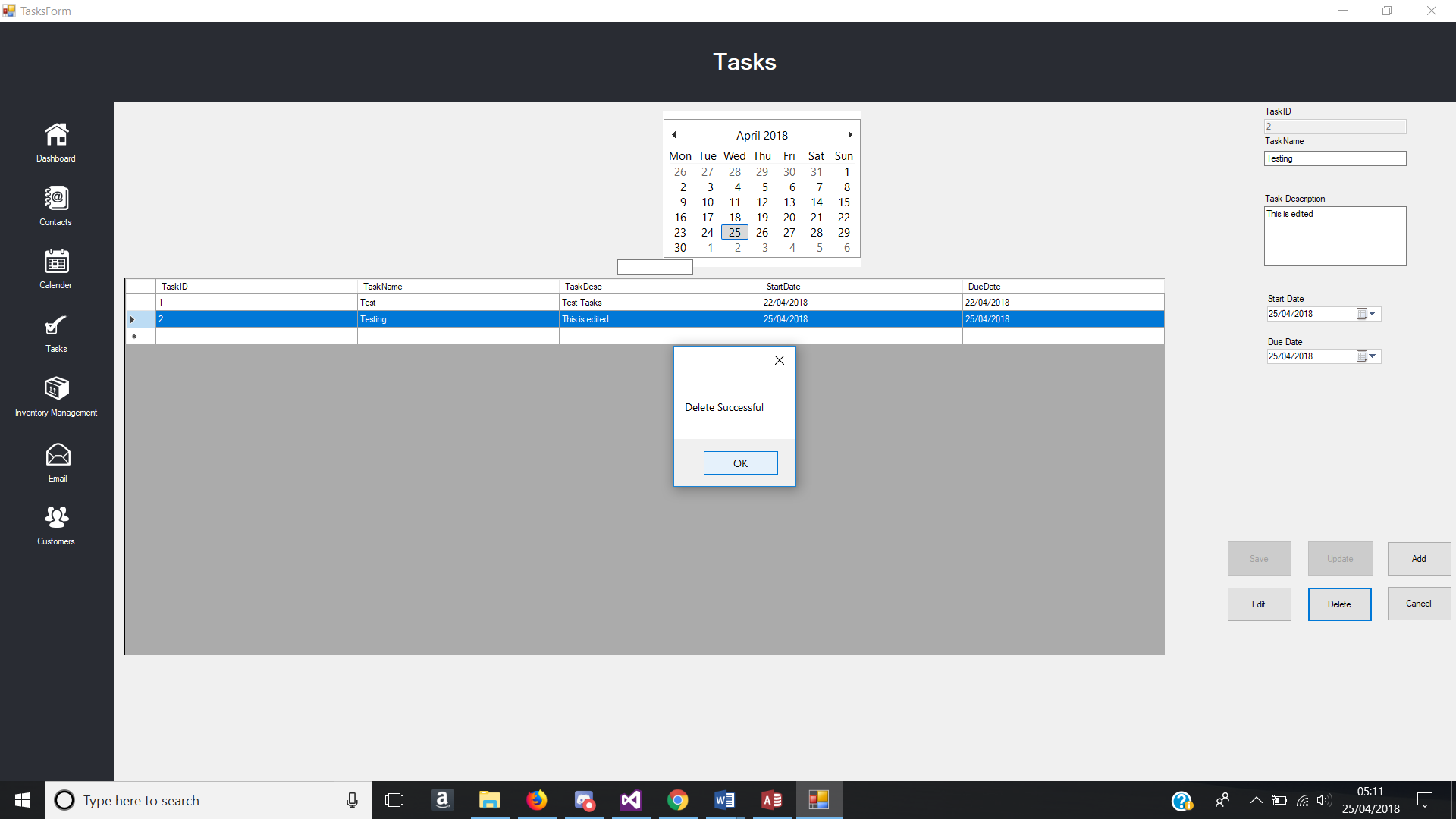


Figure - Delete Button Pressed Test

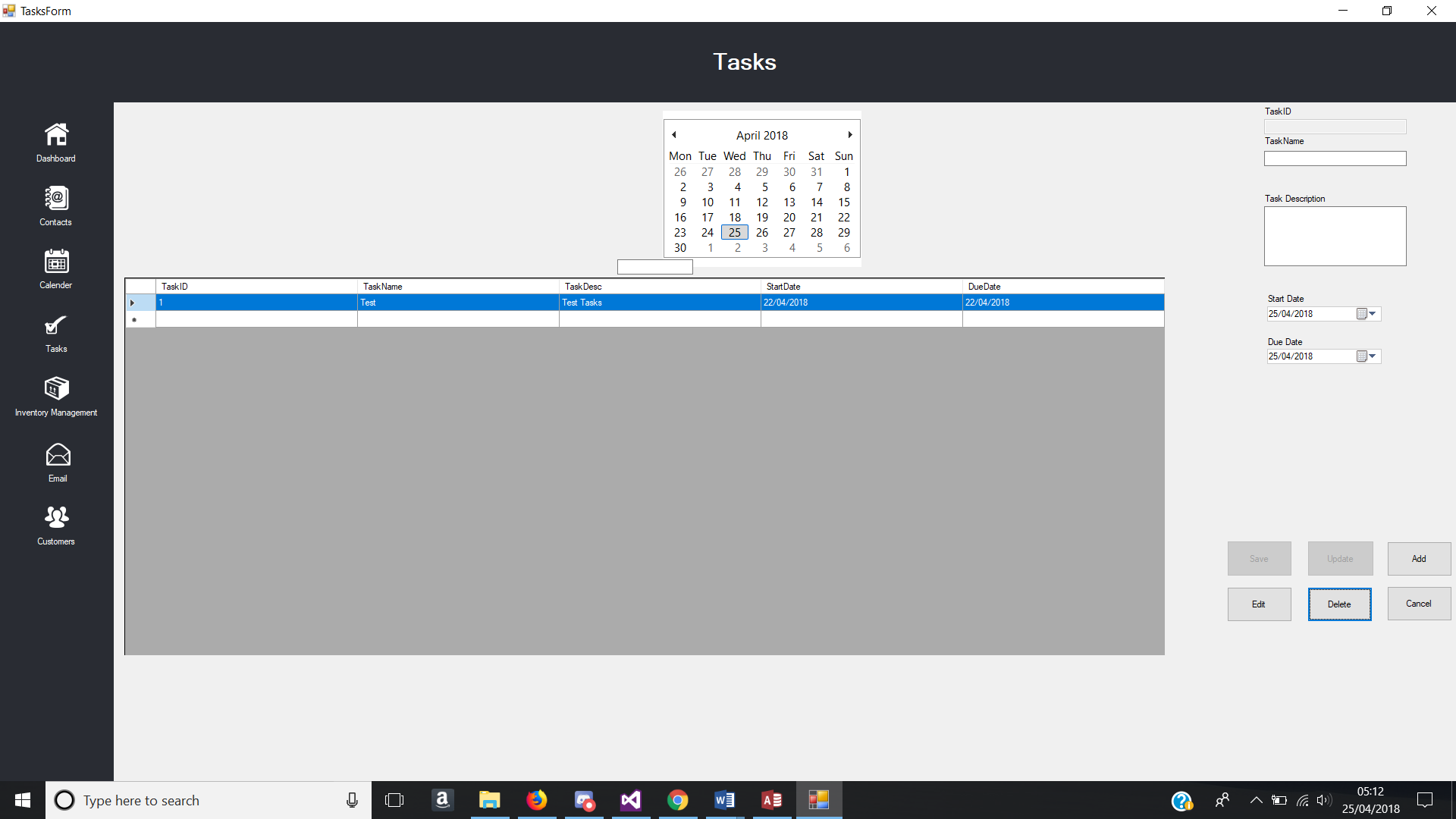


Figure - Record Deleted in data grid view

### Inventory Management Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Inventory Management Form** | | | | |
| **Test No.** | **Test** | **Event** | **Expected Result** | **Actual Result** |
| 1 | Database | Form load and database file automatically stores into local directory | Database file downloads to local directory of the system. | Same as expected. |
| 2 | Database | Database uploads to google drive | Local Database file uploads to google drive when button pressed and overwrites old file | Creates a new file in google drive instead of replacing the old file. |
| 3 | Contacts button pressed | Form changes to contacts form when contacts button pressed | Contacts Form successfully loads | Same as expected. |
| 4 | Calendar button pressed | Form changes to contacts form when contacts button pressed | Calendar Form successfully loads | Same as expected. |
| 5 | Tasks button pressed | Form changes to contacts form when contacts button pressed | Tasks Form successfully loads | Same as expected. |
| 6 | Dashboard Button press | Form changes to contacts form when contacts button pressed | Dashboard Form successfully loads | Same as expected. |
| 7 | Email button pressed | Email Form Loads | Email Form successfully loads | Same as expected. |
| 8 | Add button pressed | Textboxes are enabled for user to input data when add button is pressed | Textboxes are enabled for user to input data and browse button enabled. | Same as expected.  See Figure 48 |
| 9 | Save button pressed | Data inputted are saved into the database when save button is pressed. | Data inputted in the textboxes and date selected in data pickers are saved into the database. | Same as expected.  See figure 49 & figure 50 |
| 10 | Edit button pressed | Textboxes enabled and save and add button are disabled | Textboxes are enabled for user to input data when edit button is pressed. Add and Save button are disabled. | Same as expected  See figure 51 |
| 11 | Update button pressed | Changed data in textboxes are saved into the same record in the database. | The edited data in textboxes are saved into the record that was selected when edit button was pressed. | Same as expected  See figure 52 |
| 12 | Delete button pressed | Removes row from database | Row of data from the database is deleted | Same as expected.  See figure 53. |
| 13 | Tabs pressed | Tab changes, showing new controls | New controls are shown when a tab is pressed. | Same as expected |
| 14 | Data grid view | Data grid view row selected, and textboxes are filled with corresponding data. | Textboxes are filled with data when data grid view row is selected | Same as expected. |
| 15 | Validation | Particular textboxes have validations | Textboxes that require numbers to be inputted will only allow integer characters to be entered. | Same as expected |
| 16 | Error catch | Input something wrong, message pops up warning user | Message box appears to inform the user, what went wrong and does not crash the system. | Same as expected |
| 17 | Form Maximised | Interface changes appropriately when maximised | All form controls move to the centre when form is maximised | Same as expected |

Table - Inventory Management Form Test Table

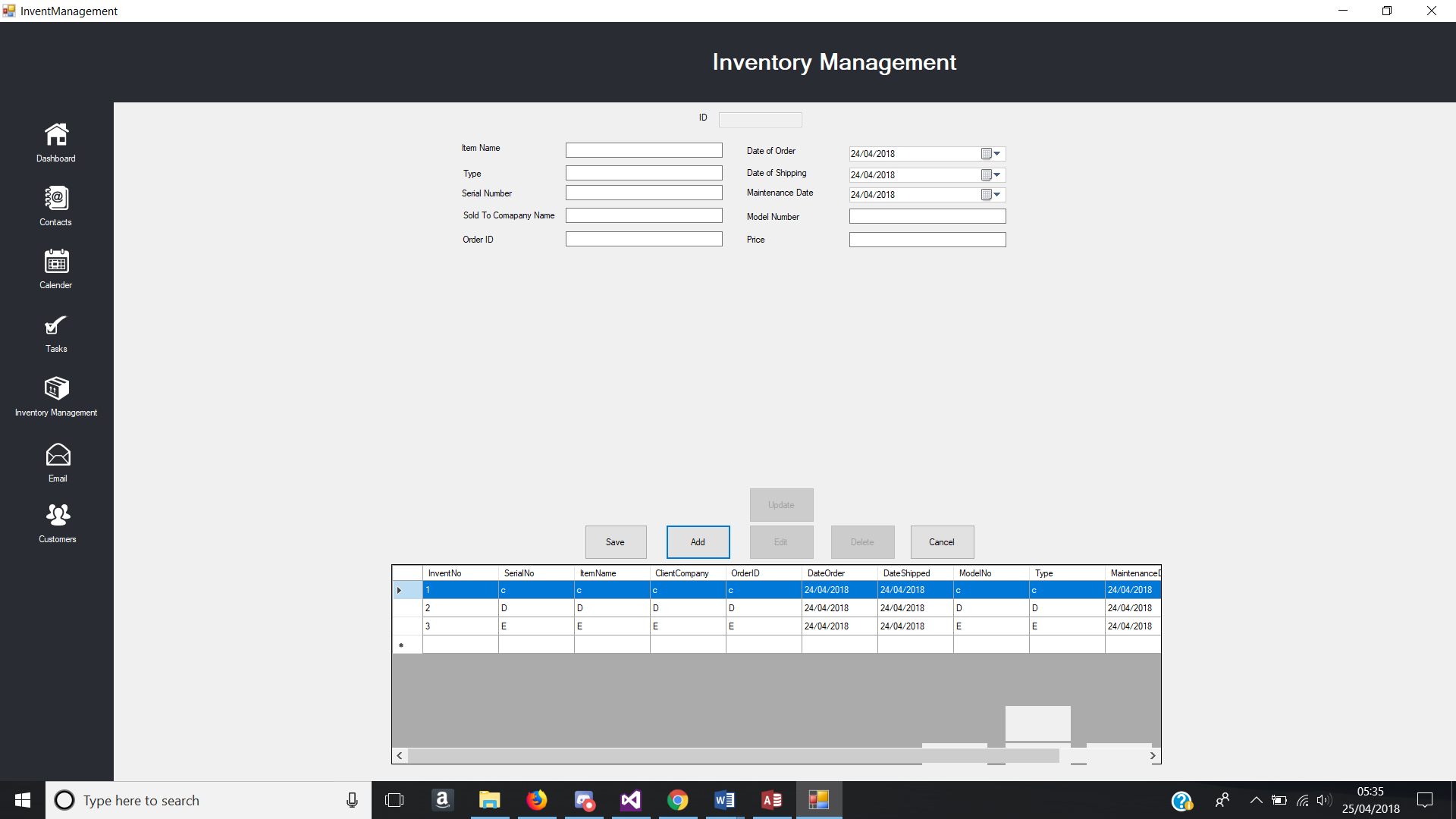


Figure - Add Button Pressed Test

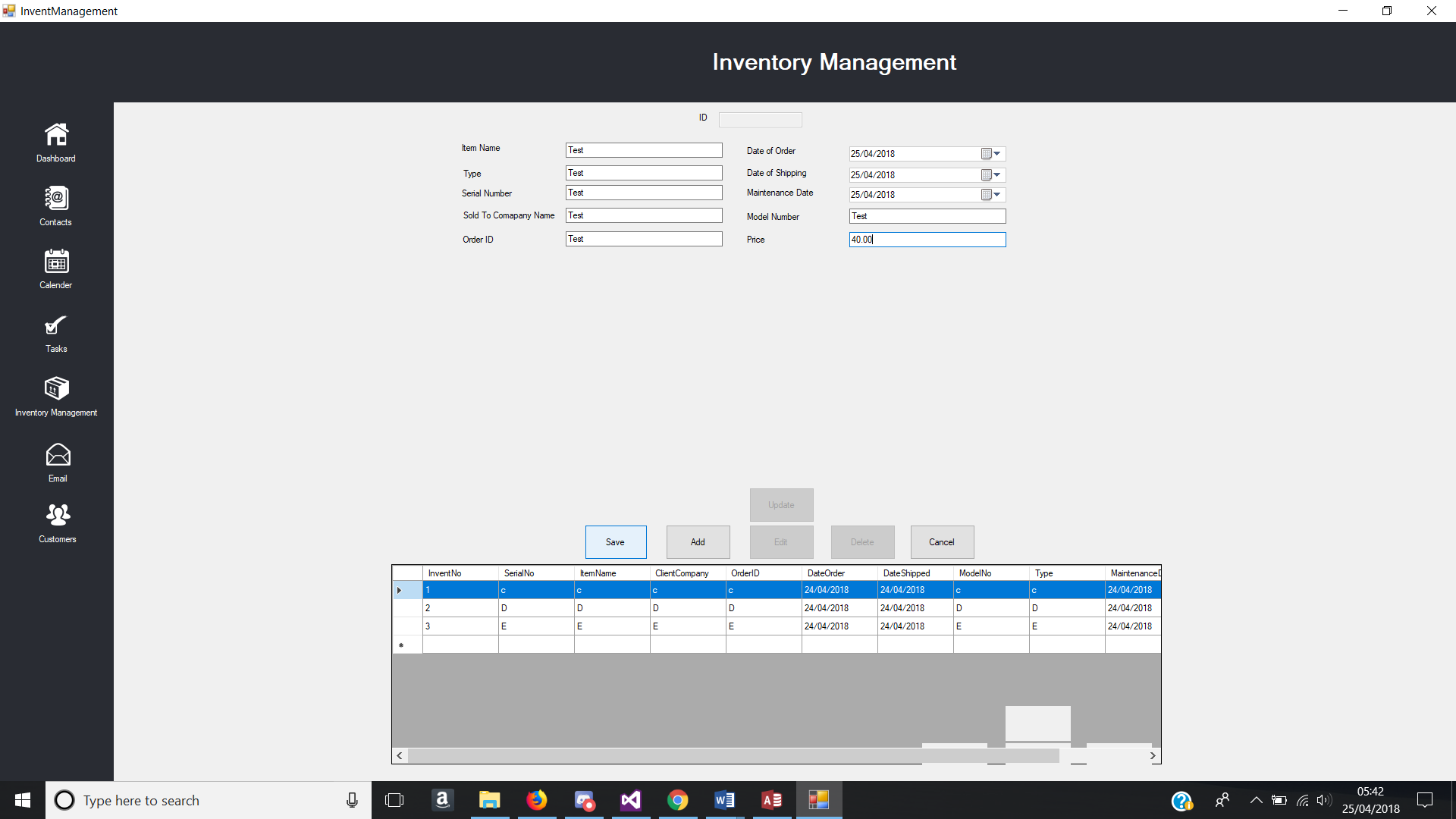


Figure - Data inputted into textbox and Dates selected

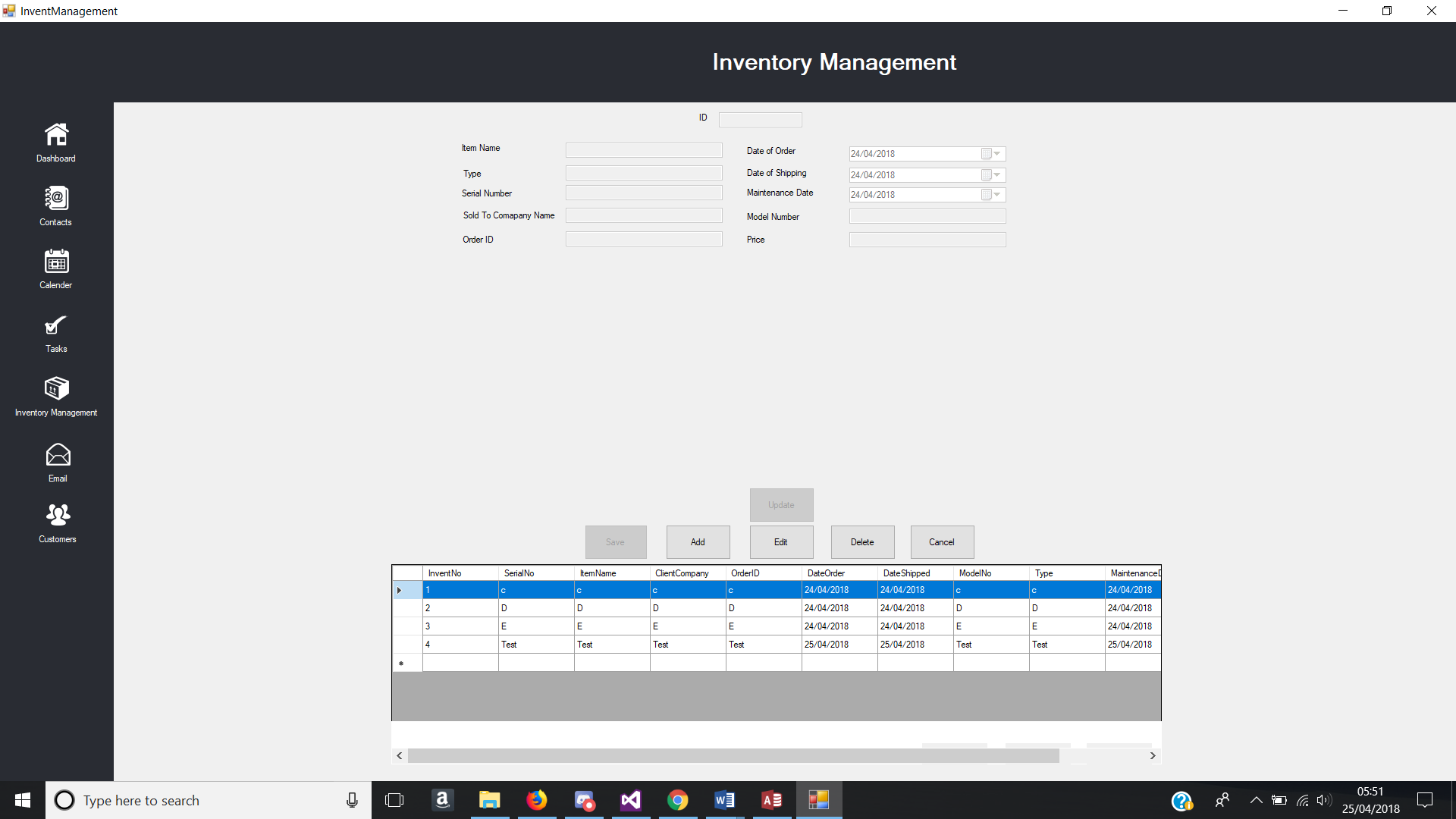


Figure - Save Button Pressed Test

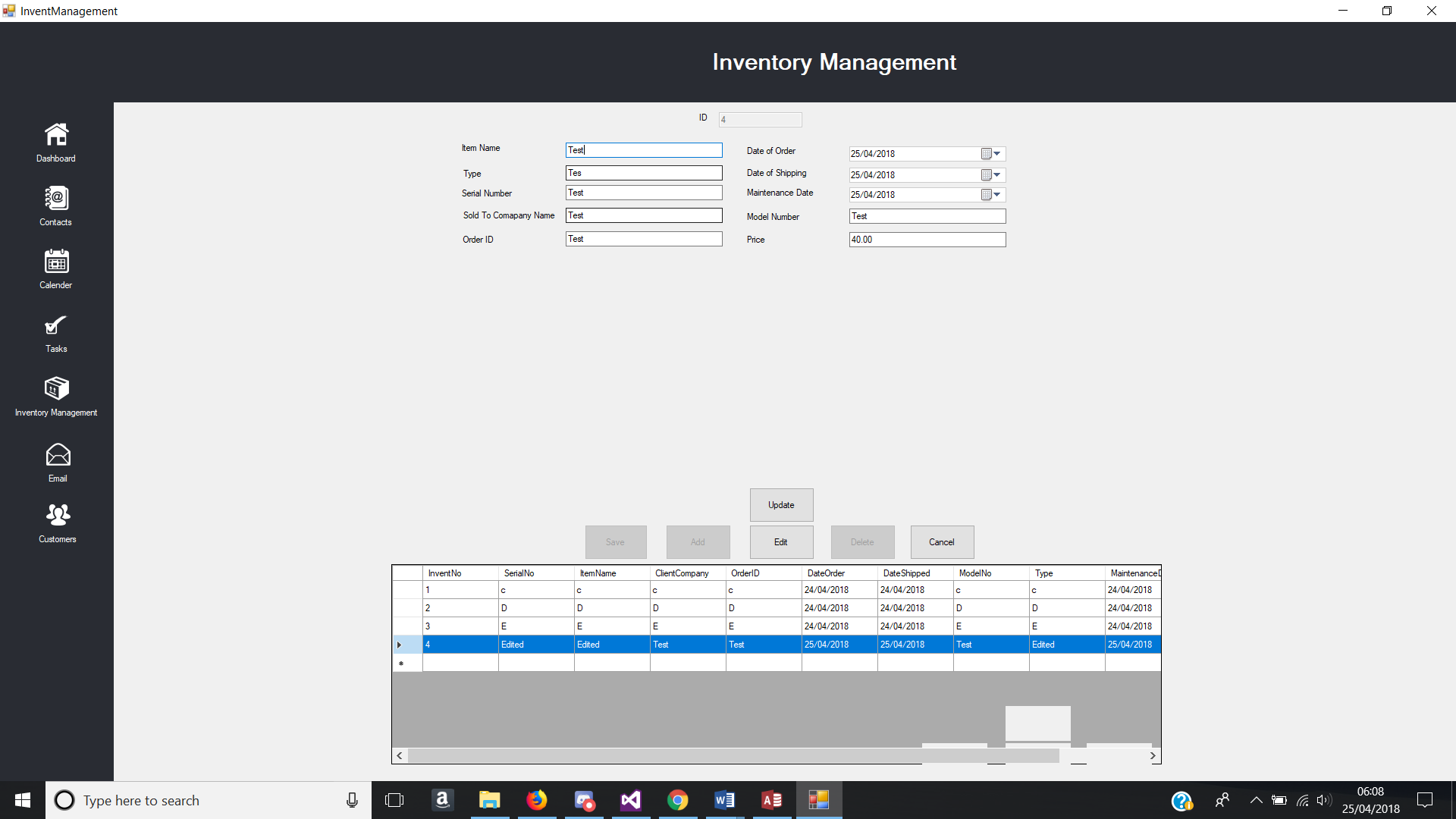


Figure - Edit Button Pressed Test

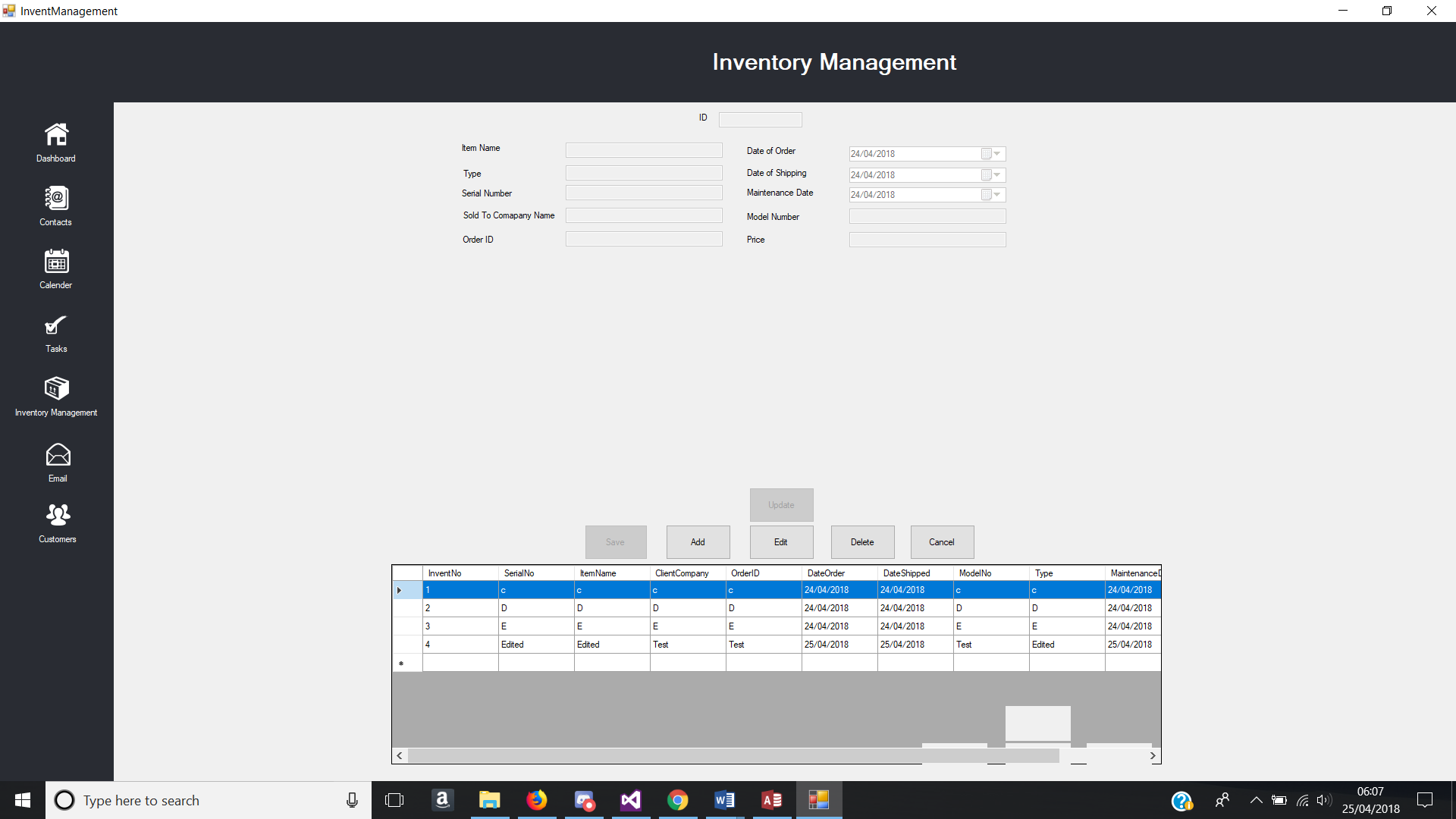


Figure - Update Button Pressed Test

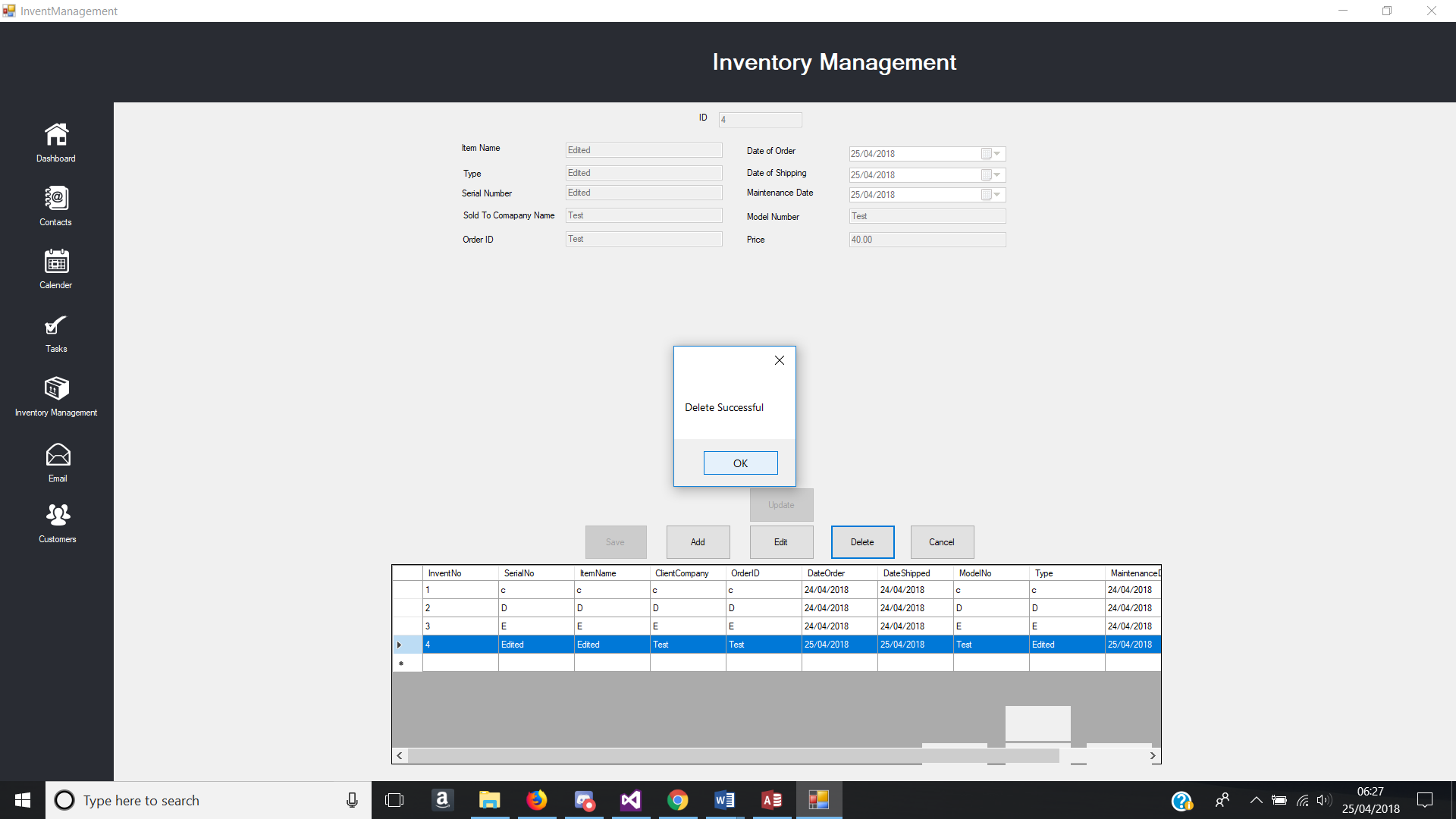


Figure - Delete Button Pressed Test

### 5.1.6. Email Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Email Form** | | | | |
| **Test No.** | **Test** | **Event** | **Expected Result** | **Actual Result** |
| 1 | Contacts button pressed | Form changes to contacts form when contacts button pressed | Contacts Form successfully loads | Same as expected. |
| 2 | Calendar button pressed | Form changes to contacts form when contacts button pressed | Calendar Form successfully loads | Same as expected. |
| 3 | Tasks button pressed | Form changes to contacts form when contacts button pressed | Tasks Form successfully loads | Same as expected. |
| 4 | Inventory Management button pressed | Form changes to contacts form when contacts button pressed | Inventory Management Form successfully loads | Same as expected. |
| 5 | Dashboard Button pressed | Form changes to contacts form when contacts button pressed | Dashboard Form successfully loads | Same as expected. |
| 6 | Email Button Pressed | Form changes to Email form when email button pressed | Email form successfully loads | Same as expected. |
| 7 | Send Button | Email sends and all empty all textboxes | Email will be sent to the attended receiver and textboxes become empty. | Same as expected. |
| 8 | Data grid view | Email inbox is shown in the data grid view | All emails are shown in the data grid view on form load and refresh button. | Same as expected  Shown in figure 54 |
| 9 | Email Content Reader | Able to read an email when clicking onto a row in the data grid view | Email contents are shown when a row is selected in the data grid view | Same as expected.  Shown in figure 55 |

Table - Email Form Test Table

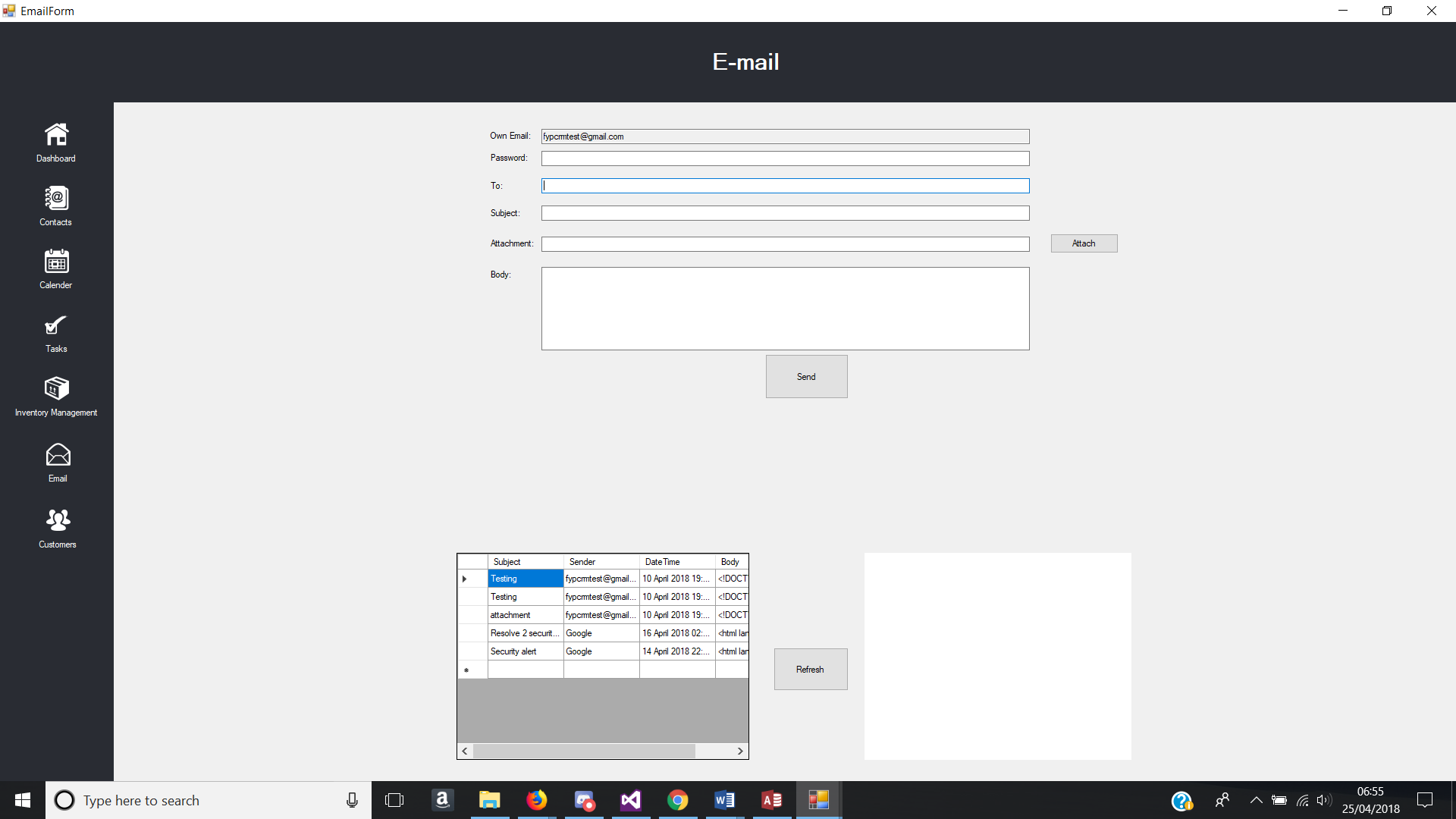


Figure - Email inbox in Data grid view



Figure - Email Content Reader Test

### 5.1.7. Customers Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Email Form** | | | | |
| **Test No.** | **Test** | **Event** | **Expected Result** | **Actual Result** |
| 1 | Contacts button pressed | Form changes to contacts form when contacts button pressed | Contacts Form successfully loads | Same as expected. |
| 2 | Calendar button pressed | Form changes to contacts form when contacts button pressed | Calendar Form successfully loads | Same as expected. |
| 3 | Tasks button pressed | Form changes to contacts form when contacts button pressed | Tasks Form successfully loads | Same as expected. |
| 4 | Inventory Management button pressed | Form changes to contacts form when contacts button pressed | Inventory Management Form successfully loads | Same as expected. |
| 5 | Dashboard Button pressed | Form changes to contacts form when contacts button pressed | Dashboard Form successfully loads | Same as expected. |
| 6 | Email Button Pressed | Form changes to Email form when Email button pressed | Email form successfully loads | Same as expected |
| 7 | Search | Filter through data and show what the has been inputted in the textbox | Data is filtered and is displayed in the data grid view | Same as expected  See figure 56 |
| 8 | Data Grid View | Data from the customers table in the database is outputted in the data grid view | Data grid view is populated with data from customers table in the database | Same as expected |

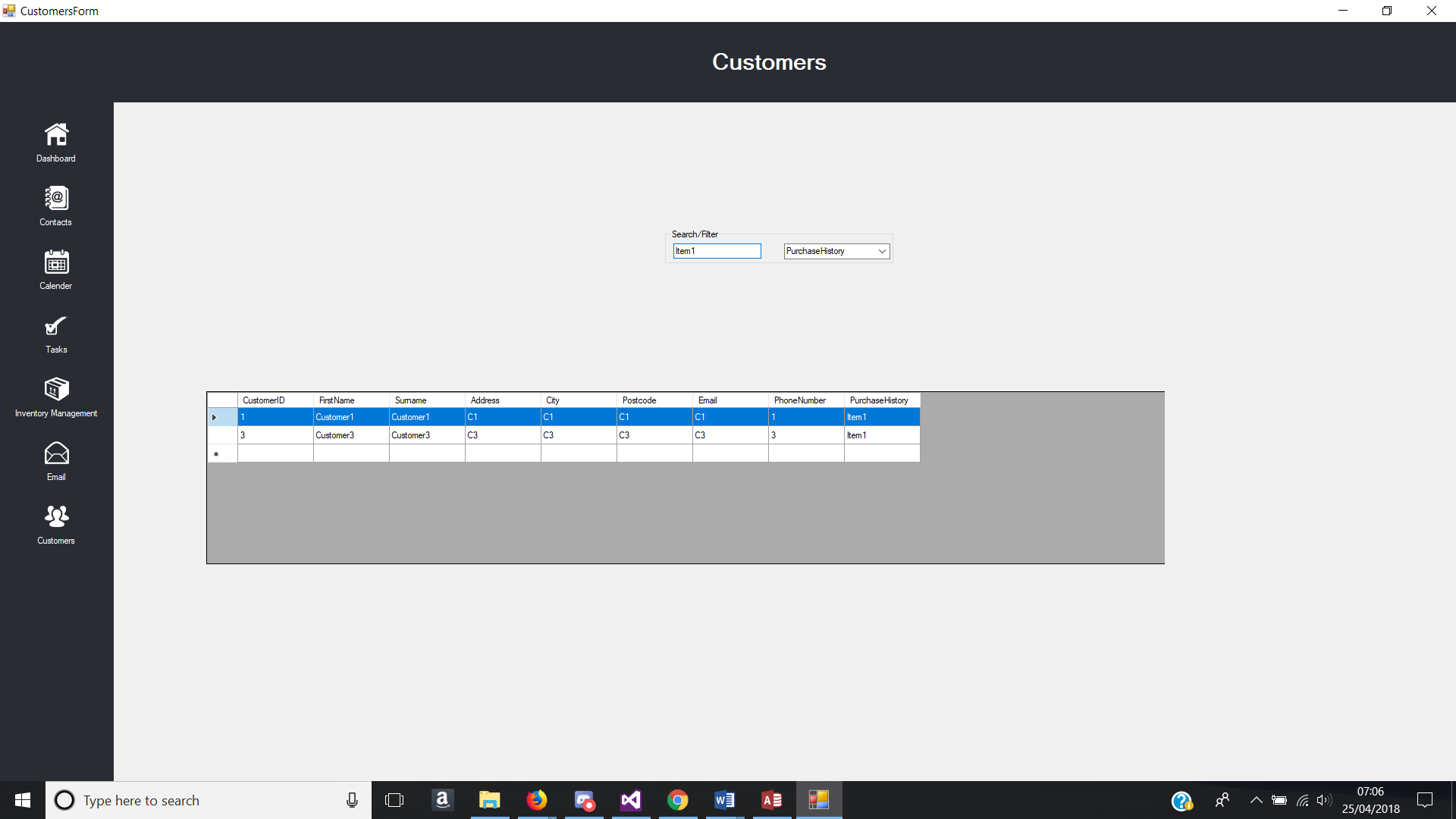


Figure - Filter Data through Search

## Implementation Issues and Challenges

There were many issues in developing the CRM. One issue is the time given to complete this project. It was difficult to ensure that everything will be completed and fully working as intended, especially implementing all the features that every CRM have and then adding the main objective of database automation, Analysis of customers, products etc. The analysis was difficult to figure which type of analysis was useful to businesses in general. Furthermore, some of the features that had hoped to be implemented as said in the proposed system could not be done, as the time frame was not enough for me to add all the implementation that was hoped to be added.

Another issue is the amount of data being used. A small amount of dummy data was used, but difficult to see how it would handle large amounts of data. Another issue this would cause is the analysis may not be 100% accurate whereas testing with 1000’s records will provide more of an accurate testing.

A major challenge that was faced during the implementation face was applying google API. Downloading and uploading to google drive which is used as a representation of cloud. Weeks were spent creating test projects to try and get the feature working before implementing onto the actual project by reading tutorials by google and trying to understand how the code works so it could be changed to suit my project. This implementation had used a majority of the time, which did not allow enough time to add other features.

# Conclusion/Future works

The reason for creating this system was to improve on current features that CRM have and implement new ideas such as automated upload and download to and from the cloud. However, the system lacks much of what current CRM systems have as there are many enhancements that could be made.

## Successful Outcomes

Starting from the research and the tests that have been conducted, it is a possibility to say that the system was a success. Some features that current CRM systems have were implemented successfully as planned and the implementation and usage of a cloud storage with a slight automation, but with that being said, this project CRM system would not compare to any of the current CRM’s out in the market as they are more sophisticated and continuously improving by adding new features.

## Unsuccessful Outcomes

The outcome of creating a fully featured CRM system that can be equal to current CRM systems out in the market was unsuccessful. Many of the implementations that were planned did not go through as some features being implemented took longer than expected with the research, implementing and testing until it works. Some features were never taught so constant research had to be conducted during the implementation phase.

## Future Enhancements

There are future enhancements that can be done for this project. The first enhancement is the features of the system. Basic core features were done, but this CRM is not fully complete and still has a lot of space to develop to a better standard with more features to be added. Automation was done in a sense of uploading and downloading from google drive, but this CRM can add SFA, SA, MA and other automations. Social media was also a big feature that was not implemented and therefore enhancing the system with social media would improve the system a lot. With these added improvements to this application, the CRM will be more complete and much more better for organisations to use.

Besides the features that could be implemented, the system can be improved by merging with other ERP systems. The system can be more adaptable to most of another ERP system which this would make this CRM adaptable to other systems and a better standalone system all at once.

Another major enhancement to this project would be having the system on a wide variety of platforms, such as web and mobile. With the project integrated with these platforms and are connected to one another, it provides different ways of accessing the system and would be useful for employees of an organisation to do work that is needed to be done no matter where and when.

# References

Dimitrova, M., 2017. Customer Relationship Automation will be the future of CRM. [Online] Available at: https://medium.com/@SoftClouds/customer-relationship-automation-will-be-the-future-of-crm-4a5735146bf3. [Accessed 30 01 2018].

Salesforce, 2004. 'CRM systems - A beginner’s guide'. [Online] Available at: https://www.salesforce.com/uk/learning-centre/crm/crm-systems/. [Accessed 6November 2017].

Lysis (2018) ‘The Disadvantages of CRM Systems’. [Online]. Available at: <https://www.techwalla.com/articles/the-disadvantages-of-crm-systems>. [accessed at 6 November 2017]

TechOneStop (N.A) 'What is Operational CRM'. [Online]. Available at: <http://techonestop.com/what-is-operational-crm>. [accessed at 20 February 2018]

Tutorialspoint (2014) 'CRM - Introduction'. [Online]. Available at: <https://www.tutorialspoint.com/customer_relationship_management/crm_introduction.htm>. [accessed at 6 November 2017]

Tutorialspoint (2014) 'CRM - Introduction'. [Online]. Available at: <https://www.tutorialspoint.com/customer_relationship_management/crm_types.htm>. [accessed at 6 November 2017]

Nibusinessinfo (N.A) 'Business benefits of CRM systems'. [Online]. Available at: <https://www.nibusinessinfo.co.uk/content/business-benefits-crm-systems>. [accessed at 6 November 2017]

Zoho (N.A) 'CRM features'. [Online]. Available at: <https://www.zoho.com/crm/features.html>. [accessed 20March 2018]

D. Prabha and R. S. Subramanian, "A survey on customer relationship management," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, 2017, pp. 1-5.

GetApp (N.A) ‘Zoho CRM vs Apptivo CRM vs Salesforce Sales Cloud Comparison Chart’. [Online]. Available at: <https://www.getapp.com/customer-management-software/a/zoho-crm/compare/apptivo-customer-app-vs-salesforce/>. [accessed at 20 March 2018]

Berglund, A. and Leifer, L., 2013. Why we prototype! An international comparison of the linkage between embedded knowledge and objective learning. *Engineering Education*, *8*(1), pp.2-15.

Hu, H.H., Hu, H.Y. and Parsa, H., 2015, August. Customer relationship management and innovation as performance drivers. In *Management of Engineering and Technology (PICMET), 2015 Portland International Conference on* (pp. 835-841). IEEE.

Patel (2017) ‘Different Types of Software Testing and How It Improves the Software Quality’ [Online]. Available at: <https://www.upwork.com/hiring/for-clients/different-types-software-testing-improves-software-quality/>. [accessed at 10 April 2018]

Prayitno, E. and Astuty, N.A., 2017, November. Positive impact of Customer Relationship Management (CRM) implementation to improving the services of animal polyclinics customers. In Sustainable Information Engineering and Technology (SIET), 2017 International Conference on (pp. 246-250). IEEE.

Shatnawi, M.Q., Yassein, M.B. and Al-natour, H., 2017, August. Customer relationship management at Jordan university of science and technology: Case study, issues and recommendations. In Engineering and Technology (ICET), 2017 International Conference on (pp. 1-6). IEEE.

Buttle, F., 2009. Customer Relationship Management: Concepts and Technology.

Brown, S.A. and Coopers, P.W., 1999. *Customer relationship management: A strategic imperative in the world of e-business*. John Wiley & Sons, Inc.

# Bibliography

Dimitrova, M., 2017. Customer Relationship Automation will be the future of CRM. [Online] Available at: https://medium.com/@SoftClouds/customer-relationship-automation-will-be-the-future-of-crm-4a5735146bf3. [Accessed 30 01 2018].

Salesforce, 2004. 'CRM systems - A beginner’s guide'. [Online] Available at: https://www.salesforce.com/uk/learning-centre/crm/crm-systems/. [Accessed 6November 2017].

Lysis (2018) ‘The Disadvantages of CRM Systems’. [Online]. Available at: <https://www.techwalla.com/articles/the-disadvantages-of-crm-systems>. [accessed at 6 November 2017]

TechOneStop (N.A) 'What is Operational CRM'. [Online]. Available at: <http://techonestop.com/what-is-operational-crm>. [accessed at 20 February 2018]

Tutorialspoint (2014) 'CRM - Introduction'. [Online]. Available at: <https://www.tutorialspoint.com/customer_relationship_management/crm_introduction.htm>. [accessed at 6 November 2017]

Tutorialspoint (2014) 'CRM - Introduction'. [Online]. Available at: <https://www.tutorialspoint.com/customer_relationship_management/crm_types.htm>. [accessed at 6 November 2017]

Nibusinessinfo (N.A) 'Business benefits of CRM systems'. [Online]. Available at: <https://www.nibusinessinfo.co.uk/content/business-benefits-crm-systems>. [accessed at 6 November 2017]

Zoho (N.A) 'CRM features'. [Online]. Available at: <https://www.zoho.com/crm/features.html>. [accessed 20March 2018]

D. Prabha and R. S. Subramanian, "A survey on customer relationship management," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, 2017, pp. 1-5.

GetApp (N.A) ‘Zoho CRM vs Apptivo CRM vs Salesforce Sales Cloud Comparison Chart’. [Online]. Available at: <https://www.getapp.com/customer-management-software/a/zoho-crm/compare/apptivo-customer-app-vs-salesforce/>. [accessed at 20 March 2018]

Berglund, A. and Leifer, L., 2013. Why we prototype! An international comparison of the linkage between embedded knowledge and objective learning. *Engineering Education*, *8*(1), pp.2-15.

Hu, H.H., Hu, H.Y. and Parsa, H., 2015, August. Customer relationship management and innovation as performance drivers. In *Management of Engineering and Technology (PICMET), 2015 Portland International Conference on* (pp. 835-841). IEEE.

Patel (2017) ‘Different Types of Software Testing and How It Improves the Software Quality’ [Online]. Available at: <https://www.upwork.com/hiring/for-clients/different-types-software-testing-improves-software-quality/>. [accessed at 10 April 2018]

Prayitno, E. and Astuty, N.A., 2017, November. Positive impact of Customer Relationship Management (CRM) implementation to improving the services of animal polyclinics customers. In Sustainable Information Engineering and Technology (SIET), 2017 International Conference on (pp. 246-250). IEEE.

Shatnawi, M.Q., Yassein, M.B. and Al-natour, H., 2017, August. Customer relationship management at Jordan university of science and technology: Case study, issues and recommendations. In Engineering and Technology (ICET), 2017 International Conference on (pp. 1-6). IEEE.

Buttle, F., 2009. Customer Relationship Management: Concepts and Technology.

Brown, S.A. and Coopers, P.W., 1999. *Customer relationship management: A strategic imperative in the world of e-business*. John Wiley & Sons, Inc.

ISTQB Exam Certification (N.A) ‘What is Incremental model- advantages, disadvantages and when to use it?’ [Online]. Available at: <http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/>