Group 5

Members: Matthew Orga, Marcus Chua, Ian Dela Cruz

Introduction

1.1 Purpose

The purpose of this SRS is to provide two-way communication between the company, Villa Rivera Wave Resort, and the developers. The SRS contains the problems, details, and solutions brought up by the client and the developers. The progress and reports made towards the development of the project are described all throughout the SRS. This document is specifically made for the company, Villa Rivera Wave Resort, and also serves as an agreement between what the company asks and what the developers have provided. The program's main target audience is the staff as they will input the information of the guest bookings into the program which will update the local database for all the staff to view for easier management.

1.2 Scope

Software Products	Function	Limitation	Application
Booking System	• Gets input (reservation, cottages, etc.) from the customers • Catch overlapping reservations	• Delete current info that was inputted • Edit submitted responses	• **Goal**: To get customer Inquiries • **Benefits**: Reduces technical/instrumental and human error seen with traditional paper-and-pen systems. Conflict resolution between reservations. •**Objectives:** Provide a visual means of inputting inquiries
Login Software	• Check whether the user is a guest or a staff • The system should direct them to the booking system or database.	• Could only recognize if the login details are a staff. • Could not verify if the user is a staff, if a guest knows a staff account	• **Goal**: To limit on who would be able to access the system • **Benefits**: Reduces unwanted access to unauthorized users • **Objective**: Assist users in accessing a certain part of the

			system, like the information and booking tabs
Database for the booking system	• Storage of guest information for booking in a SQL file • Filter the information in set categories (ie. Booked or not booked, cottage or not) • Should allow staff to edit • Should have a search function	• It may mix up the position of the booking information • staff edit causing bugs and errors	• **Goal:** Should help the staff in maintaining the booking of cottages • **Benefit:** should help in checking the status of booking • **Objective:** To store the information of the guest booking.

Table 1.2.1 Scope

The programming languages to be used in developing the software are Java and MySQL. The study focuses on the integration of the software products mentioned above into one system. Furthermore, the system only utilizes a local database, meaning it can only be accessed on one device.

1.3 Definitions, Acronyms, and Abbreviations

CSV - CSV stands for comma-separated values (BigCommerce, 2022). It is mostly seen in excel spreadsheets and google sheets. It formats the data inputted into a tabular format for easier access.

Database - It is a stored collection of data in a computer system (Oracle, n.d.).

GUI - GUI stands for Graphical User Interface (Stoltzfus, 2021). This is what users will see as they open applications that users could interact with, using buttons, graphics, etc.

RAM - “Random access memory(RAM) is a computer’s short term memory, which it uses to handle all active tasks and apps”(Villinger, 2021).

JDK17 - used in building applications with the Java programming language

1.4 References

Oracle. (n.d.). What is a database. Retrieved April 7, 2022 from https://www.oracle.com/ph/database/what-is-database

Stoltzfuz, J. (2021, May 28). Graphical user interface (GUI). Techopedia. Retrieved April 7, 2022 from https://www.techopedia.com/definition/5435/graphical-user-interface-gui

What is a .CSV file and what does it mean for my ecommerce business? BigCommerce. (2022, March 30). Retrieved March 31, 2022, from https://www.bigcommerce.com/ecommerce-answers/what-csv-file-and-what-does-it-mean-my-ecommerce-business/

Villinger, S. (2021, August 5). What is Ram and why is it important? Retrieved April 7, 2022, from https://www.avast.com/c-what-is-ram-memory

1.5 Overview

Chapter 1 of the SRS provides a brief background of the SRS, together with its purpose, scope, and intended audience.
Chapter 2 provides the overall product description, containing the different features, interfaces, accessibilities and requirements needed for the product.
Chapter 3 contains the specific software details of the product, such as the necessary inputs, outputs and the processes performed by the software.
Chapter 4 contains the change management process.
The activity diagrams consist of representations of the different activities and classes in the program.

Chapter 2: The Overall Description

Overall description

The chapter discusses the different features, interfaces, and functions of the product. It also provides the limitations and constraints of the product and the technical and physical requirements needed to use the product. Lastly, it provides future requirements that can be addressed in future versions of the product.

2.1 Product Perspective

The product will be used to solve inquiry and reservation problems experienced by the customer. It would therefore function as a booking or logging system that would be utilized by the company. A similar or existing product to this system is a logbook that uses a traditional pen and paper system, which is the current method being used by the company. In comparison to the traditional method, the product would provide more accuracy to the user or company when it comes to booking as the reservations are documented properly, thereby lessening human error and technical complications.

Figure 2.1.1 Product Perspective

2.1.1 System Interfaces

The product is dependent on MySQL workbench, an application that will be utilized for storing the information of the inquiries, and it will be used as the database. Aside from that, it is independent of other applications as the software would is hard-coded by the developers

2.1.2 Interfaces

The *Booking System* will be using a GUI for both the customer inquiry as input and the database as output. The GUI for the booking system will use a form-like interface, similar to what can be seen in google forms, for the customer inquiry. While the database GUI will have a priority system, for example, the name is arranged by alphabet or putting the reserved cottages at the top, for easier accessibility to the data.

2.1.3 Hardware interfaces

The product can be accessed using a computer or a desktop. This system will use a keyboard in order to input the data needed for the logbook system that will be created, and a mouse for navigation controls.

2.1.4 Software Interface

The system's local input phase will be using Java to create the GUI application and should run with Windows 10. The software specifications in which the system phase is built are:

(1) Java
(2) JDK 17

The database storing and displaying will be using Java.

2.1.5 Communication interface

The system will not require an internet connection as it uses a local-based database.

2.1.6 Memory Constraint

The product's use of memory would not exceed 500Mb of RAM.

2.1.7 Operations

The system should have a filtering system to help the categorizing of the data, such as the reservation of cottages or a different housing area, once the inputted one is complete. The database will also have a search engine in order for the searching of a certain user to be easier.

2.1.8 Site adaptation Requirements

Ideally, the system should be able to adapt to many different operating systems in order to make it more accessible to different kinds of hardware.

2.2 Product functions

The system is divided into two phases; customer information input, and the storing and displaying of the database. For the customer input, a GUI application will be used to systematically select booking variables such as customer name, contact details, cottages, amount of people, and time of visit. Upon input, the product checks whether there are complications such as overlapping reservations and missing information. If there are no input issues, then database storage will commence where the information will be logged into the databases.

2.3 User characteristics

The system will be accessible to users with a high school and above educational level since the interface will have a form-like structure like google forms. The form will contain business terms, specifically for reserving in a resort, so having an educational level below high school can cause confusion in interacting with the system.

2.4 Constraints

The statement on the data privacy act will also be stated in the forms since it will be needed in the customer's booking information. The system should not be able to send the information of the customer to other sources other than the local database.

2.5 Assumption and dependencies

The system's input phase requires only Windows 10 as an operating system. However, the system's data storing and displaying phase require programming languages that are compatible with most modern hardware.

2.6 Apportioning of requirements

Currently, there are no requirements that are in need of reprioritization nor updates to be implemented.

Chapter 3: Specific Requirements

3.1 External Interfaces

3.1.1 Login Software

Username and password

- Its output would result in either a success or failure. It is a string variable that is used for the login system of the program. It is used for identifying if the user is a staff member or not. The user only has a certain amount of tries to log in with the correct username and password. The item is dependent on the database and crosschecks the values if it corresponds to a certain value in the database. A correct login would lead to the main menu of the software.

3.1.2 Booking System

Customer Name

- This stores the name of the customer who is reserved for a particular time in the resort. Upon entry and crosschecking together with the other information, it will be added to the database. A complete entry would save the values to the database.

Cottage

- Its purpose is to reserve the customer under a certain cottage. It is an input value together with the other information that is added to the database. The item is one of the values that would be crosschecked in the database on whether the chosen cottage is already reserved or not.

Date and time

- Its purpose is to reserve customers on a given date and time. It is another input value together with the other information added to the database. The date and time have a tolerance within 14 days of the entry. Similarly, it is one of the values that will be crosschecked in the database on whether the selected time has another reservation already or not.

Amount of guests

- It is a numerical input on the number of guests who would enter the resort, based on the inquiry of the customer. Its main purpose is to check whether the number of guests mentioned by the customer would fit inside the cottages and whether the total number of guests would exceed the capacity of the resort on that specific day. If it succeeds, the information is then inputted into the database together with the other values and the customer has booked their reservation.

Room

- Rooms are similar to the cottages in which it is an optional part of the inquiry. Whenever a customer applies for a room, it is cross-checked with the database whether the desired room is available for that particular day.

3.1.3 Database

Data

- This interface stores the output that is collected from the booking system. It will output a list of the inputted information that was retrieved from the booking system in a sorted manner. The sortation of the information will be based on the time slot that the customer reserved for that particular day.

3.2 Functions

The login system shall permit valid users and reject incorrect attempts through a collection of valid accounts (username-password pairs). It shall direct the user to the main menu once login has been validated as true, and will prompt users otherwise.

The booking system shall perform crosschecking to prevent overlapping inquiries and reservations between users. It shall check whether the date, time and cottage do not match with the other inputs from the database. It also checks whether the number of guests does not exceed the limit for both the cottage and the resort. The system shall show errors whenever such matches were detected from the input. Upon complete verification, the information provided would then be added to the database.

The database shall sort the information obtained from the booking system in a chronological arrangement. It shall also have the function of editing the information for the possibility of the customer backing out of their reservation.

3.3 Performance Requirements

Static Numerical Requirements

The system could only handle one user at a time since it is a local system. The type of information that would be mainly handled by the system is integer and string data types. These values contain the customer and reservation information that would be stored in the database. The system only supports one terminal as it is a local project software.

Dynamic Numerical Requirements

The system's database crosschecking shall be processed within 2 seconds.

3.4 Logical Database Requirements

The database includes the following information to be processed: (1) Customer name, (2) Cottage, (3) Date and Time, (4) Amount of guests, and (5) Rooms. The database will be accessed whenever a new inquiry was inputted, or when general viewing of the database information is initiated. Only staff and managers who have access to a company account will have access to the database. Further, data retention will only be retained for 14 days past the day of reservation.

3.5 Design Constraints

The system is only available for one terminal which could affect productivity in the booking of customers. The accessibility of the software is limited as well since it will be a local database and can not be accessed outside of the original terminal.

3.5.1 Standards Compliance

The system is not under any legal or standard constraints. The database is self-sufficient and independent from other networks and software.

3.6 Software System Attributes

3.6.1 Reliability

The system is independent of other software and does not rely on external modules. Since it is a local database, it does not rely on internet connectivity and can work offline. The system mainly relies on the terminal or the computer where the software is installed.

3.6.2 Availability

The system will run only on-demand or when accessed. The users can restart the system after failure with the loss of the currently inputted information, the information that was inputted on the previous run of the system will still be available.

3.6.3 Security

Currently, the login system consisting of the username-password pair, is the main type of security for the program. It is best that new login credentials are utilized each day to minimize potential breaches.

3.6.4 Maintainability

The system is a local project which does not feature external features or internet connection. Therefore there is little to no maintenance required for the system to continuously operate. The maintenance that may occur is due to the update of the operating system of the device or an update of the application used to develop the system.

3.6.5 Portability

The system software being local, is 100% host-dependent. Currently, the program will also run best if not only, on Windows 10 operating system.

3.7 Organizing the Specific Requirements

3.7.6 Response

The system will act like a logbook that outputs the response of the user into a database and will sort the information. Specific responses included are:

- A correct input of username and password would lead to the booking system
- An incorrect response would prompt the user to input a correct account
- A unique input of information in the booking system would reserve it in the database
- A non-unique input of information in the booking system would prompt the user to change a certain value from the input

3.8 Additional Comments

Currently, there is no other organizational technique that is ideal for the software.

Chapter 4: Change Management Process

In following up changes for the software, a series of steps or processes must first be followed by both client and customers. These steps are as follows:

4.1 Creating a request for change

Request for Change Process	
Client-request	**Developer-request**
- A client can request changes by using the platforms; Email (gmail), where a request for a scheduled Zoom meeting is also possible. - Change suggestions will be discussed with the developers. - The client should first mention to the developers the changes they want to perform.	- Developers would use a different platform for communication whenever a request for change is going to be created. - Change suggestions will be discussed and finalized amongst the Developers. - The developers should first mention to the client the changes they want to perform.

Table 4.1.1 Request for Change Process

4.2 Reviewing and assessing a request for change

Upon receiving the request, the change will be reviewed based on skill and technical sides. For skill-based, the change requested should be within the capabilities of the developers. Otherwise, the change could be referred to an external developer who is more experienced in handling that specific request. In terms of technicality, the change request should only be within the scope of the agreed requirements. Furthermore, the change should only be applied to the produced software and will not require the developers to create another or external product. This review and assessment would ensure that the change requested are achievable before proceeding with planning and implementing the change.

4.3 Planning the change

The change will be checked whether it can be added into the code and how to add the code without ruining the code as a whole. The code should still do what it is intended to do after the change is applied, so it is thought out in this part of the sequence. Also, the change should be approved by the developers and the client, to ensure that both parties' needs are met.

4.4 Testing the change

Multiple planned tests would have to be done before implementation. This includes the different testing methods, for instance, unit testing then integration testing followed by regression testing. A modified version of the program will then be deployed for acceptance testing. Importantly, schedules and deadlines would have to be made and met to efficiently progress.

4.4.1 Unit Testing

This tests individual and interrelated modules/functions within the program.

4.4.2 Integration Testing

This would examine different groups of functions (as one) once the modification is integrated into the program.

4.4.3 Regression Testing

This test would ensure that upon completion of the modification, the whole software still runs optimally, efficiently and correctly.

4.4.4 Acceptance Testing

Finally, the client would test and assess the modified software for validation.

4.5 Creating a change proposal

The previous change processes will be documented and submitted to the client for further checking and approval by them to check whether the change satisfies the client’s order. The proposal would contain details of the requested changes, suggested solutions and the results of the tested changes. This will be reviewed by both the client and developers first before implementing the requested changes.

4.6 Implementing changes

Once the proposal has been accepted by the clients, the changes will be implemented to the code. Using the proposal details on how to implement the changes as a guide in adding and even testing the changes.

4.7 Reviewing change performance

Upon deployment of the software, different factors would have to assessed both by developers and client. For instance, an increase in runtime would be expected but must be kept at a reasonable and acceptable range. From the client, reviews may include the functionality, and ease-of-use of the modification made. Further requests from the client would require the process stated in section 4.1.

4.8 Closing the process

After implementation and review, the product can is now operational. The SRS would be updated so that it meets the current requirements of the product. The changes made are taken note of and the previous version of the software may be stored by the developers. The proposal and documents would also be stored. The change process is officially done once the product is operational at the working site.

UML ACTIVITY DIAGRAM

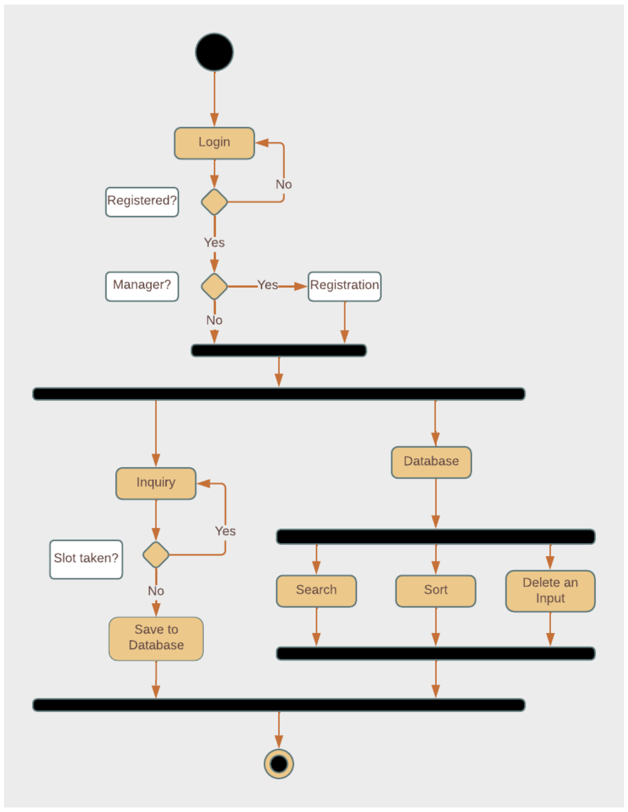


Figure 1. UML Activity Diagram

Overview:

At the start of the program, the user will be asked to log in to the system. If the user is not a registered user, the system will re-prompt the user to log in again. Else, an option whether the user would like to register another account to the system or not.

After logging in, the user could either go to inquiry or database. For the inquiry, the user will input the needed information to be placed in the database. If the entered information is similar to an already existing entry or slot, they will be asked to inquire again, else, the entry is saved to the database. For the database, the user could either search, sort, or delete a specific entry in the database.

UML Class Diagram

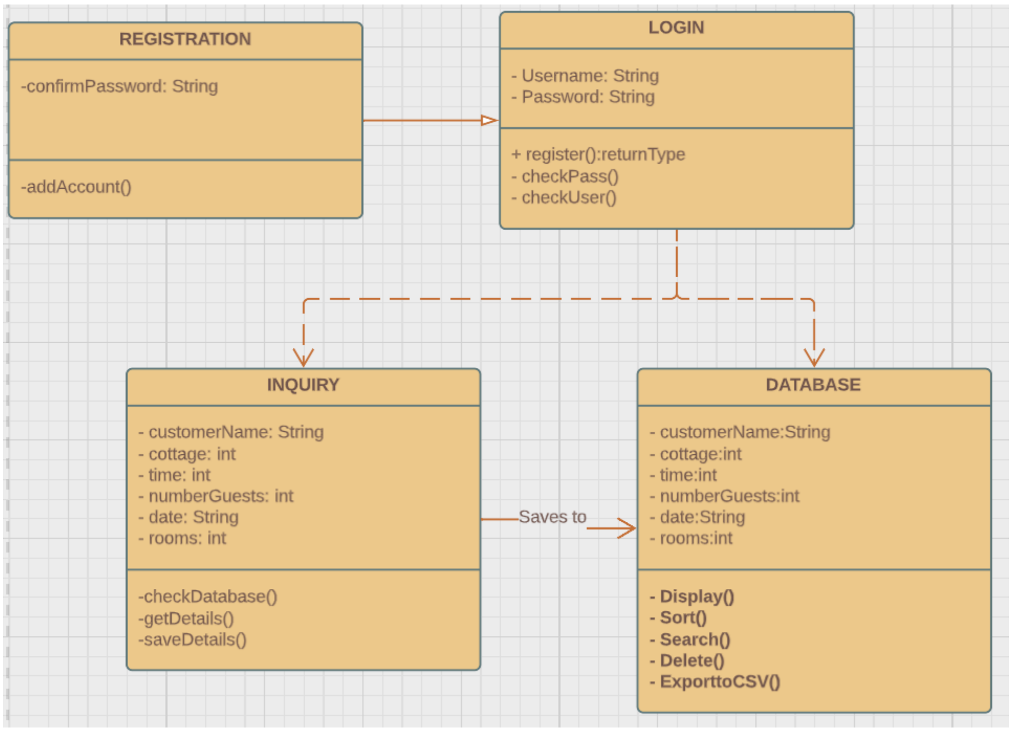


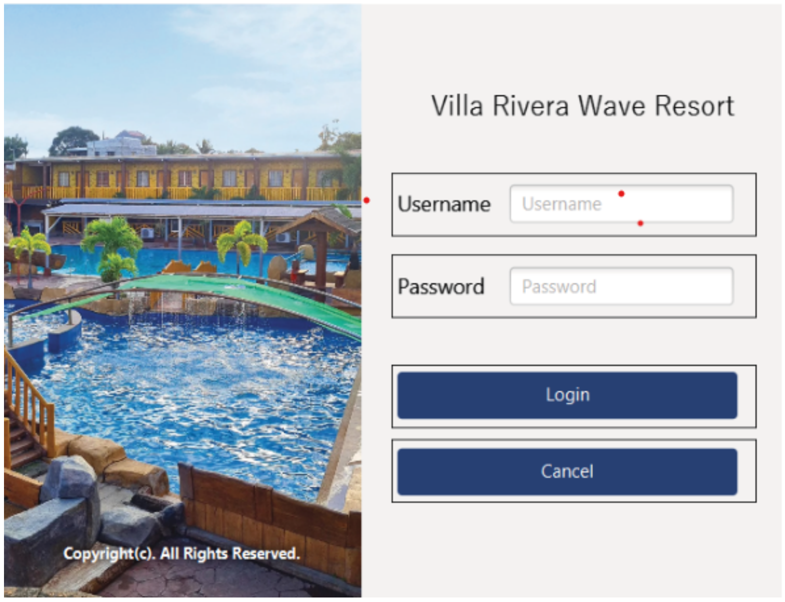
Figure 2. UML Class Diagram

Overview:

The system's classes are identified on Figure 2. The Login class contains the different attributes such as the *string* credentials needed for access, Username and Password. Methods include verifying the inputs and registering a user. The Inquiry class contains the information as attributes needed for each reservation. Methods are for cross validation, getting the inputs and saving the inputs. Lastly, the Database class contains the same attributes and methods are for displaying, sorting, searching, deleting, and exporting saved data.

Appendix

Login Program Software User Manual



Villa Rivera Wave Resort

Username [Username]

Password [Password]

Login

Cancel

Copyright(c). All Rights Reserved.

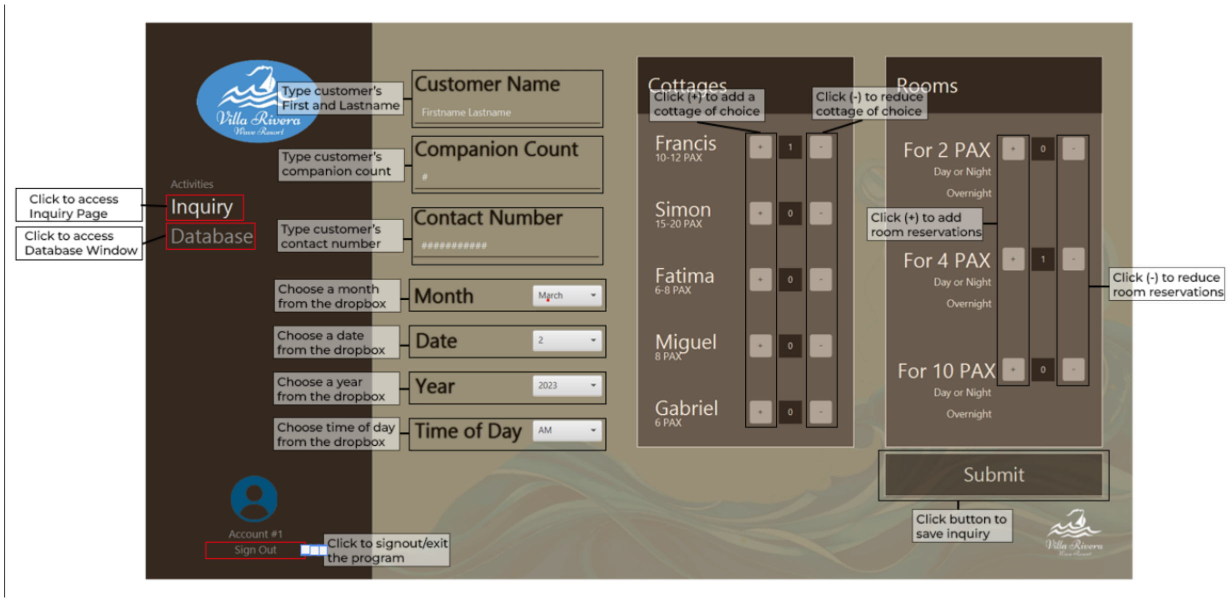
Type your Username ①

Type your Password ②

Click the Login Button ③

Click the Cancel Button to exit ④

Inquiry Program



Click to access Inquiry Page — Inquiry

Click to access Database Window — Database

Type customer's First and Lastname — Customer Name [Firstname Lastname]

Type customer's companion count — Companion Count

Type customer's contact number — Contact Number [###########]

Choose a month from the dropbox — Month [March]

Choose a date from the dropbox — Date [2]

Choose a year from the dropbox — Year [2023]

Choose time of day from the dropbox — Time of Day [AM]

Account #1

Sign Out — Click to signout/exit the program

Cottages — Click (+) to add a cottage of choice; Click (-) to reduce cottage of choice

	+		-
Francis 10-12 PAX	+	1	-
Simon 15-20 PAX	+	0	-
Fatima 6-8 PAX	+	0	-
Miguel 8 PAX	+	0	-
Gabriel 6 PAX	+	0	-

Rooms — Click (+) to add room reservations; Click (-) to reduce room reservations

	+		-
For 2 PAX Day or Night Overnight	+	0	-
For 4 PAX Day or Night Overnight	+	1	-
For 10 PAX Day or Night Overnight	+	0	-

Submit — Click button to save inquiry