|  |
| --- |
| MS6: On The Square Customer Reward Program |
| Business Buccaneers: William Kozel (Systems Analyst), Johann Quintero (Business Analyst), Jack Fentke (Programming Coordinator), Paige Lowery (Project Manager) |

****

|  |
| --- |
| 12-6-2017 |

Table of Contents

[Project Description 3](#_Toc500352337)

[Executive Background 3](#_Toc500352338)

[Current Business Environment 4](#_Toc500352339)

[Problems with Current System 4](#_Toc500352340)

[Proposed System Objectives and Constraints 5](#_Toc500352341)

[Business Benefits 5](#_Toc500352342)

[Stakeholders 5](#_Toc500352343)

[Analysis 6](#_Toc500352344)

[Use Case Package 6](#_Toc500352345)

[Executive Summary 6](#_Toc500352346)

[Context Diagram 6](#_Toc500352347)

[Use Case Diagram 7](#_Toc500352348)

[Use Case Specifications 8](#_Toc500352349)

[Use Case – 1 (Login) 8](#_Toc500352350)

[Use Case – 2 (Manage Account) 9](#_Toc500352351)

[Use Case – 3 (Manage Point Balance) 10](#_Toc500352352)

[Use Case – 4 (Generate Report) 13](#_Toc500352353)

[Use Case – 5 (Manage Redeemable Rewards) 14](#_Toc500352354)

[Use Case – 6 (Manage Guest Accounts) 15](#_Toc500352355)

[Specifications 16](#_Toc500352356)

[Functional Requirements 16](#_Toc500352357)

[Non-Functional Requirements 16](#_Toc500352358)

[Hardware/Software Specifications: 17](#_Toc500352359)

[Standard Naming Convention 17](#_Toc500352360)

[Design 18](#_Toc500352361)

[Complete Data Flow Diagram Package 18](#_Toc500352362)

[Context Diagram 18](#_Toc500352363)

[Data Flow Diagram Level 0 18](#_Toc500352364)

[Data Flow Diagram – Level 1 19](#_Toc500352365)

[Entity Relationship Diagram 21](#_Toc500352366)

[Navigation Diagram 22](#_Toc500352367)

[Program Plan 23](#_Toc500352368)

[Mock Forms 32](#_Toc500352369)

[Login 32](#_Toc500352370)

[Staff Menu 33](#_Toc500352371)

[Add New Guest 34](#_Toc500352372)

[Manage Guests/Rewards 35](#_Toc500352373)

[Owner Login 36](#_Toc500352374)

[Owner Menu 37](#_Toc500352375)

[Owner Guest Management 38](#_Toc500352376)

[Owner Wine Management 39](#_Toc500352377)

[Generate Reports 40](#_Toc500352378)

[Sample Reports 41](#_Toc500352379)

[Structured Walkthrough 43](#_Toc500352380)

[Challenges, Problems, and Discoveries – Test Plans 43](#_Toc500352381)

[Program Code 43](#_Toc500352382)

[Login 43](#_Toc500352383)

[Landing 44](#_Toc500352384)

[New Guest 44](#_Toc500352385)

[Add Guest Points 46](#_Toc500352386)

[Edit Existing Guests 48](#_Toc500352387)

[Redeem Guest Points 51](#_Toc500352388)

[Owner Login 52](#_Toc500352389)

[Owner Management 53](#_Toc500352390)

[Manage Guest Profiles 53](#_Toc500352391)

[Select Report 54](#_Toc500352392)

[Test Data 56](#_Toc500352393)

[Training 57](#_Toc500352394)

[Staff Training Documentation 57](#_Toc500352395)

[All Staff Functions: 57](#_Toc500352396)

[Login Screen 57](#_Toc500352397)

[New Guest Information Window 59](#_Toc500352398)

[Existing Guests Window 60](#_Toc500352399)

[Owner Only Functions 61](#_Toc500352400)

[Owner Login 61](#_Toc500352401)

[Owner Management Window 62](#_Toc500352402)

[Manage Guest Window 63](#_Toc500352403)

[Manage Wine Specials Window 64](#_Toc500352404)

[Report Selection Screen 65](#_Toc500352405)

[Technical Manual 66](#_Toc500352406)

[Database 66](#_Toc500352407)

[The Wine Database 66](#_Toc500352408)

[Future Growth 67](#_Toc500352409)

[Trouble Shooting 67](#_Toc500352410)

[Forms 68](#_Toc500352411)

[Manage Wine Specials form: 75](#_Toc500352412)

[Project Assessment 77](#_Toc500352413)

[Johann Quintero (Business Analyst) 77](#_Toc500352414)

[Jack Fentzke (Database Coordinator) 77](#_Toc500352415)

[William Kozel (Systems Analyst 78](#_Toc500352416)

[Paige Lowery (Project Manager) 78](#_Toc500352417)

[Project Plan 79](#_Toc500352418)

[Appendix 81](#_Toc500352419)

[Accomplishments 81](#_Toc500352420)

[Problems Encountered 81](#_Toc500352421)

[Timesheets 81](#_Toc500352422)

# Project Description

## Executive Background

The following document for On The Square’s Information System includes the history of the business and explains business methods currently being utilized and our plans for the new system. Our team will provide information about our proposed system and its expected benefits for the company. The document will also provide other requirements needed to build this information system including functional and nonfunctional requirements as well as context diagrams, and our full use case package.

On The Square opened their doors in October 2003 as a small tapas bar and lounge in Tarboro, North Carolina. The restaurant was already not following the typical southern dining practices that most people were used to in Eastern NC; but when they received such success from the start of the business, they decided to take it a step further and transform the restaurant into a “quintessential urban bistro, without the urban”. Now, almost 14 years later, they are serving dishes with seasonal and sustainable ingredients from all over NC, and local fishermen for their seafood. On The Square has attracted patrons from all over the United States, but they also maintain a strong local customer base. Although the leading reason they have been able to keep their doors open is the amazing food; another thing that keeps customers coming back and recommending this restaurant is the southern hospitality that they receive from the minute they walk through the door, until they leave. The restaurant is open Monday through Friday for lunch, which is a more casual dining experience, and they open for dinner Thursday through Saturday for a fine dining atmosphere.

Based on our conceptual walk-through in Milestone 1, we have decided as a team to stick to a web based project that will revamp the entire restaurants current website. We will create a site that will allow customers to do all of the projected features that we presented in our first Milestone.

## Current Business Environment

On The Square is a small business that is independently owned and operated by husband and wife, Stephen and Inez Ribustillo. Stephen is the executive chef and creates and designs all of the items on the menu, while Inez is a Master Sommelier and runs the retails wine room in the front of the restaurant. There are around 30 employees that rotate working each week. The staff includes 2 full-time managers, a kitchen manager and a dining room manager, that work directly with Stephen and Inez to make sure that the restaurant is ran properly. On The Square is a small fine dining restaurant that utilizes local superior ingredients to elevate classic southern dishes. Although the Ribustillo family runs a successful small business, there is room for improvements. Their customer base for the most part is locals that live within a 30-mile radius of Tarboro, North Carolina; any out of town customers are usually first time diners or do not come back often. There is no incentive to get customers coming back to the restaurant and there is little to no customer relations program.

## Problems with Current System

The current systems for customer relations consist of a VIP system through email. The system works by allowing customers to become VIP after using their email to book a certain amount of reservations. The customers become VIP in the system but nothing extra is given to them because of this, making them the same as any other customer. Without a proper customer relations system, the company will never be able to fully develop a beneficial relationship with their customers.

As of right now, the VIP program is the only thing On The Square utilizes to create a customer program. There is no loyalty program and nothing that allows customers to get updates about promotions or new menu items.

## Proposed System Objectives and Constraints

The objective of this system is to increase customer loyalty and to spark excitement in loyal, and infrequent, patrons of On the Square through a CRM. Our CRM will promote new menu items, offer discounts, and offer points to loyalty members who continue to patron our store. This will boost and bolster current sales figures and increase store profits. This will help On the Square grow from a small local eatery to a larger regional location that people will drive from miles around to enjoy. Additionally, we want to better display On the Square and what they have for offer to potential customers. In terms of skill and total ability.

Our constraints are our team size, On the Square’s budget, On the Square’s schedule, On the Square’s dedication to new technological systems that are new and different, and our team’s coding experience. We believe that these are goals we can work through and overcome.

## Business Benefits

A CRM gives the ability to collect and manage customer information to see their past visits and preferences. The use of this system can give On The Square an option to send coupons and rewards to customers individually as part of a customer loyalty program. This will allow more attention to each customer and potentially result in more frequent visits to the business. This kind of program can allow customers to interact with the business and increase the diversity of customers to more than just locals.

## Stakeholders

The primary stakeholders are those who are directly involved with On The Square. This includes the owner, the employees who work there, and the customers who patron the restaurant. Other stakeholders include other local restaurants that are considered the restaurant’s competitors

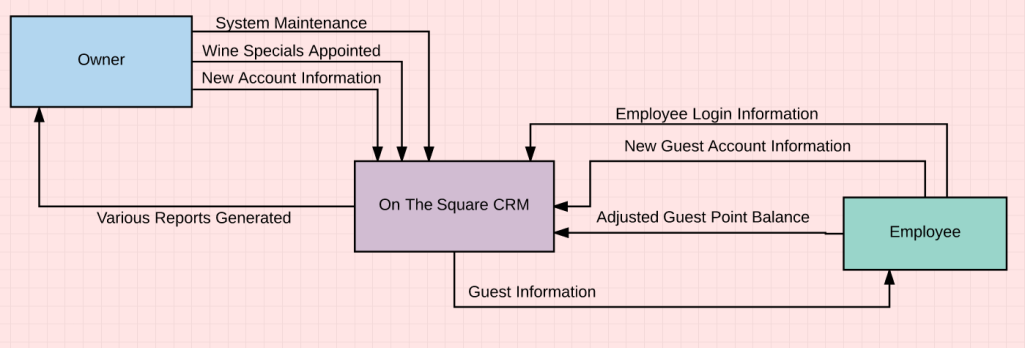
# Analysis

## Use Case Package

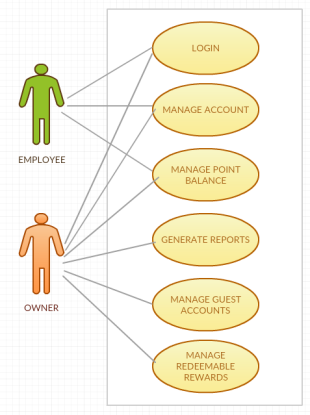
### Executive Summary

This is a comprehensive analysis of On The Square restaurant’s information system that concentrates on determining and evaluating the requirements for the system. Included in this analysis is a context diagram of On The Square’s information system, which is a high level view of the proposed system that defines the boundaries between the system, part of the system, and its environment, showing the entities that interact with it and the data flow between them. Also the analysis contains the use case diagram of the system that identifies, clarifies, and organizes the requirements and who interacts with their system including actors, use cases, and supporting actors. The use case specifications are analyzed last in the document and highlight the requirements for how a user will interact with the system to achieve value. This document should outline the requirements, actors, preconditions, flow of events, normal flow, and alternative flows of each use case for On The Square’s new proposed system.

### Context Diagram



### Use Case Diagram



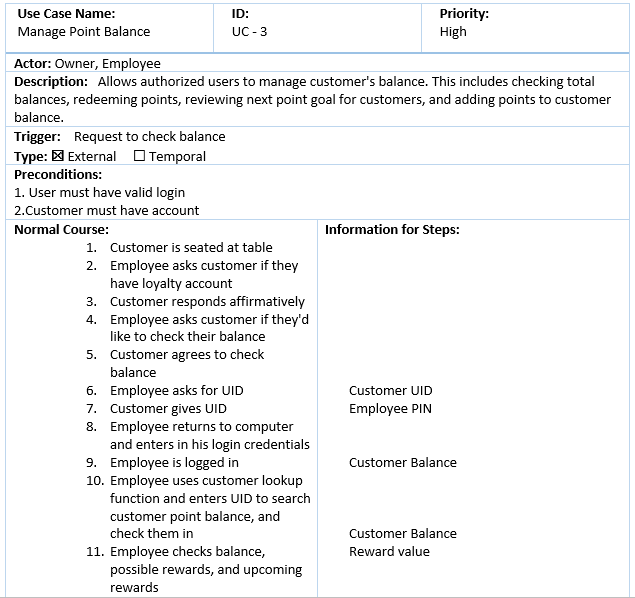
### Use Case Specifications

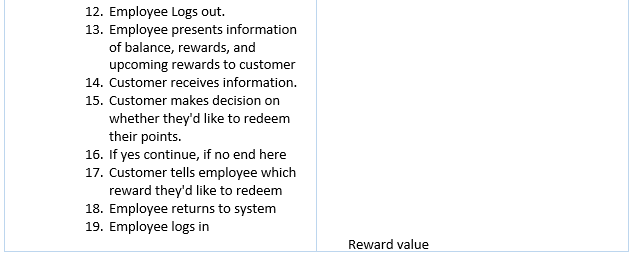
Use Case – 1 (Login)

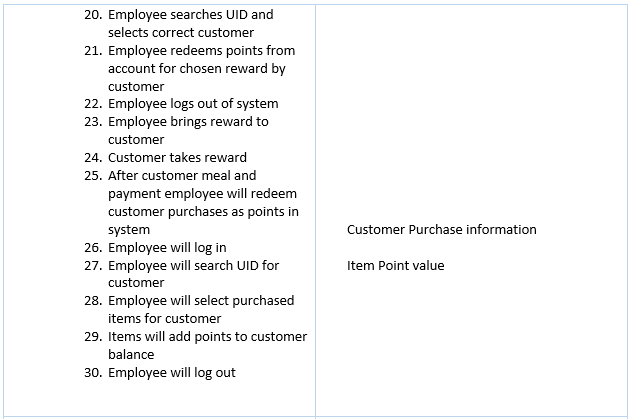
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case Name:  Login | ID:  UC - 1 | | Priority:  High | |
| Actor: Staff (Owner/Employees) | | | | |
| Description: Allows the actor to gain access to the system and the information specific to that actor | | | | |
| Trigger: User login  Type: ☒ External ☐ Temporal | | | | |
| Preconditions: Staff has PIN assigned by Owner | | | | |
| Normal Course:   1. User attempts to log into the system 2. Enters their assigned PIN 3. System accepts request to access  * If User has Owner login they will be granted access to more menu options that are not privy to employees  1. User is now logged in | | | | **Information for Steps:**    PIN (Five Digits) User access granted |
| Alternative Courses:   1. Staff enters the wrong PIN    1. Prompted to Try Again | | **Information for Alt. Course:**  PIN (Five Digits) | | |
| Postconditions:  1. Owner has access to complete system  2. Staff can view and manage all information other than reports and specific item point values | | | | |
| Exceptions:  Staff PIN is Pre-Existing  Owner will always have access to entire system   |  |  |  |  | | --- | --- | --- | --- | | Summary Inputs:  PIN (Five Digits) | Source:  Staff | Outputs:  User access granted | Destination:  Staff/CRM | | | | | |

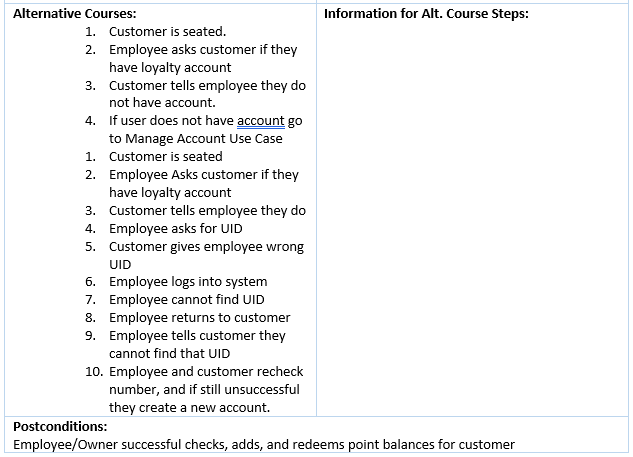
### Use Case – 2 (Manage Account)

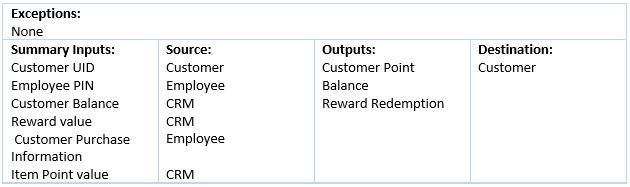
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case Name:  Manage Account | | ID:  UC - 2 | | Priority:  High | |
| Actor: Staff (Employee or Owner) | | | | | |
| Description: This use case is for the staff to create a Guest Account | | | | | |
| Trigger: New Guest Account needs to be created on-site.  Type: ☒ External ☐ Temporal | | | | | |
| Preconditions:  Staff has a valid login and acquired Guest Information | | | | | |
| Normal Course:   1. Staff logs into CRM 2. Selects "Create Guest Account" 3. The Staff adds a new Guest to the CRM with their information. 4. The CRM asks for confirmation 5. Staff confirms the account creation 6. New Account created 7. Staff logs off | | | **Information for Steps:**  Staff log in information  Guest information, Name, ZIP, Phone Number, DOB, Email  New Guest Account | | |
| Alternative Courses:   1. Staff selects the wrong button 2. Staff goes to previous screen and selects "Create Guest Account" | | | **Information for Alt. Course Steps:** | | |
| Postconditions:  New Guest Account is created | | | | | |
| Exceptions:  None | | | | | |
| Summary Inputs:  Staff Log on  Guest Name, ZIP, Phone Number, DOB, Email | **Source:**  Staff  Guest | | **Outputs:**  New Guest Account | | **Destination:**  Data Store |

Use Case – 3 (Manage Point Balance)

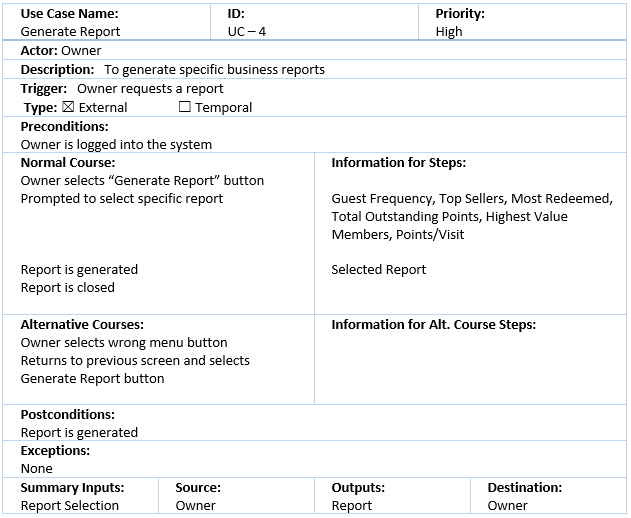








### Use Case – 4 (Generate Report)



### Use Case – 5 (Manage Redeemable Rewards)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case Name:  Manage Redeemable Rewards | | ID:  UC-5 | | Priority:  Medium | |
| Actor: Owner | | | | | |
| Description: Create and delete redeemable rewards Guests can claim based on point value system | | | | | |
| Trigger:  Type: ☒ External ☐ Temporal | | | | | |
| Preconditions:   1. User has correct credentials 2. Owner sees need to update redeemable items 3. Owner is logged into system | | | | | |
| Normal Course:   1. Owner selects "Reward Options" 2. Owner edits and manages new and old rewards, assigns points, and checks number of redemptions. 3. Owner saves and closes window 4. Owner logs out | | | **Information for Steps:**  New reward options  Updated reward options | | |
| Alternative Courses:  1. User forgets to save options before logging out  2. Window asks user if he'd like to save before exiting   1. Owner tries to save duplicate reward 2. System warns user item already exists | | | **Information for Alt. Course Steps:**  Warning Message  Duplicate reward information  Warning Message | | |
| Postconditions:  System reward options are updated | | | | | |
| Exceptions: | | | | | |
| Summary Inputs:  New Reward Options  Duplicate reward information | **Source:**  Owner  Owner | | **Outputs:**  Updated Reward Options  Warning Message | | **Destination:**  CRM  Owner |

### Use Case – 6 (Manage Guest Accounts)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case Name:  Manage Guest Accounts | | ID:  UC-6 | | Priority:  Medium | |
| Actor: Owner | | | | | |
| Description: Assign point values to new and old menu items for guests to earn. | | | | | |
| Trigger:  Type: ☒ External ☐ Temporal | | | | | |
| Preconditions:   1. Owner is logged into the system 2. New Menu Items to be added | | | | | |
| Normal Course:   1. New Item is added to physical menu 2. Owner selects "Item Values" option 3. Owner selects "New Item" 4. Owner fills out information about new menu item 5. Owner adds decided point value to menu item 6. Owner saves item 7. Owner logs out | | | **Information for Steps:**  New Menu Item  New Menu Item information  Item Point Value  New CRM menu item and point value | | |
| Alternative Courses:   1. User enters item already existing 2. System asks user if they'd like to update existing item 3. User selects yes or no to update new Item. | | | **Information for Alt. Course Steps:**  Old Menu Item  Old Menu Item Info | | |
| Postconditions:  Menu Item value is updated and assigned | | | | | |
| Exceptions: | | | | | |
| Summary Inputs:  New Menu Item Information  New Menu Item  Item Value | **Source:**  Restaurant  Owner  Owner | | **Outputs:**  New Item  New Item Point value  Old Menu Item Information | | **Destination:**  CRM  CRM  CRM |

## Specifications

### Functional Requirements

· The system must run on C# and Access (Software our team is comfortable working in)

· The system must have the ability to store local data

· The system must be able to take user credentials to avoid fraudulent entries

· The system must be able to produce reports for benefit of the managers

· The system must be able to recognize incomplete entries and prompt for completion

· The system must run on Windows 7 or higher

· The system must allow users to edit records and add new users

### Non-Functional Requirements

· The system shall run on Chrome, Firefox, Safari and Internet Explorer

· The system shall be able to run on iOS 8 or later

· The system shall be able to run on Android KitKat or later

· The system shall be available at 99.9% uptime

· The system shall be able to download full database backups daily

|  |  |
| --- | --- |
| Reliability | High user testing to ensure uptime during heavy business hours |
| Serviceability | Needs to be easily worked on so that the user can correct errors |
| Scalable | Must be able to accept large number of menu items, guests, and reward items |
| Usable | System shall be easily understood by staff with little knowledge of CRMs |
| Secure | System shall be able to easily perform backups of data |
| Modern | System shall be able to run on current PC hardware |

When we are building the system for On The Square we need to be sure that we are creating a system they can really use, and one that will meet their daily needs and requirements. The system must maintain a **reliable** characteristic, the user shouldn’t expect the system to shut down during peak hours, or cut out on them at pivotal high traffic moments of the day. During the rare moments that the user may experience an error it should be easily fixed and **serviced** within the architecture software, meaning system downtime shouldn’t be expected to go for over 40 minutes if an error occurs. For the majority of the time though we want our system to be running, and while it’s running we want our users to actually **use** the system and have good GUI experience. And after their use we want them to be able to backup and **secure** the data they entered so they can expect to see it in the same place the next day. We are also requiring this system to run on **modern** hardware and software so that in the case our client does want to **up scale** the system.

### Hardware/Software Specifications:

The software for is being designed in C# so the computer will need the following specs based on C# software requirements, database editing requirements, and preliminary knowledge from the team’s experience with C# based programs.

Software:

• Microsoft Windows 10

• Microsoft .NET Framework 4.6 (Or Higher)

• Microsoft Access

Hardware:

• Core i5 processor

• 1920X1080 monitor

• DVI video output

• USB Keyboard & Mouse

• Ethernet connection

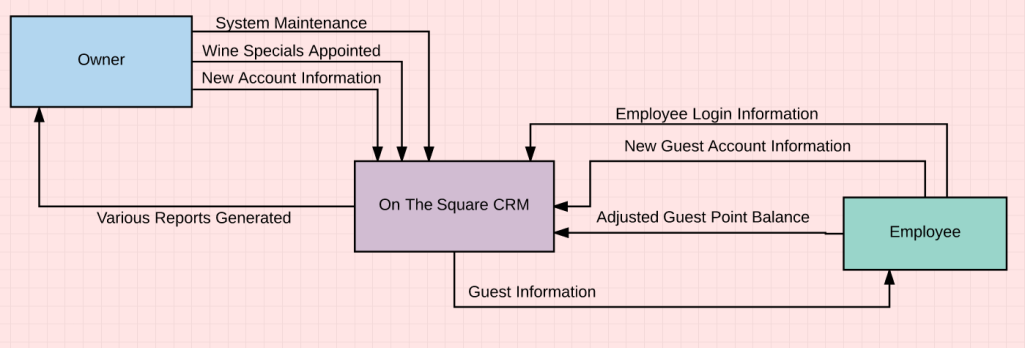
### Standard Naming Convention

* Standard C# Naming

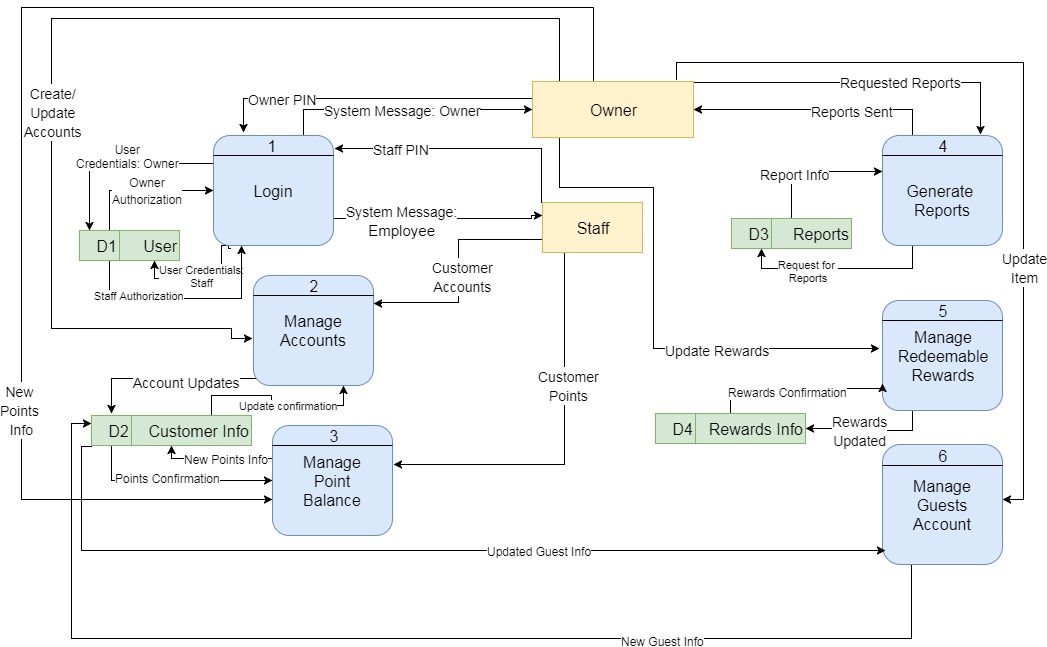
# Design

## Complete Data Flow Diagram Package

### Context Diagram

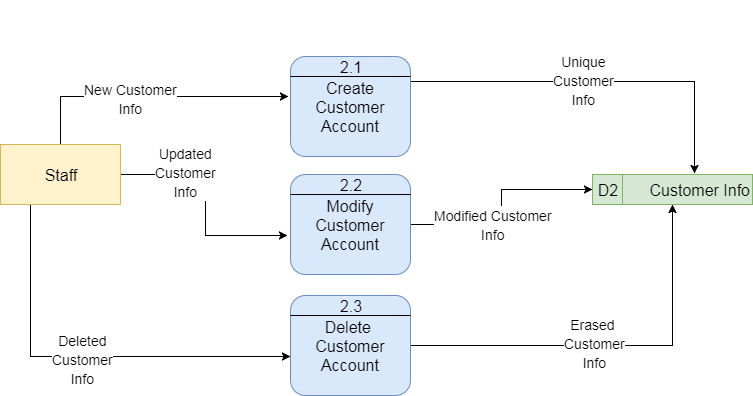


### Data Flow Diagram Level 0

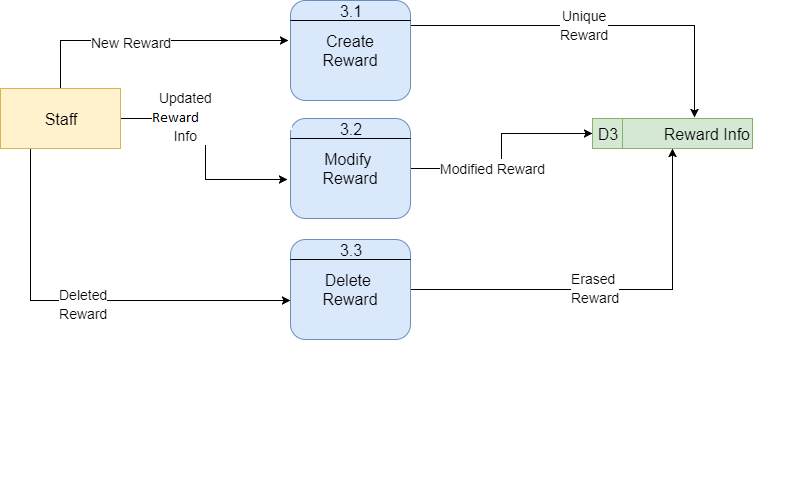


### Data Flow Diagram – Level 1

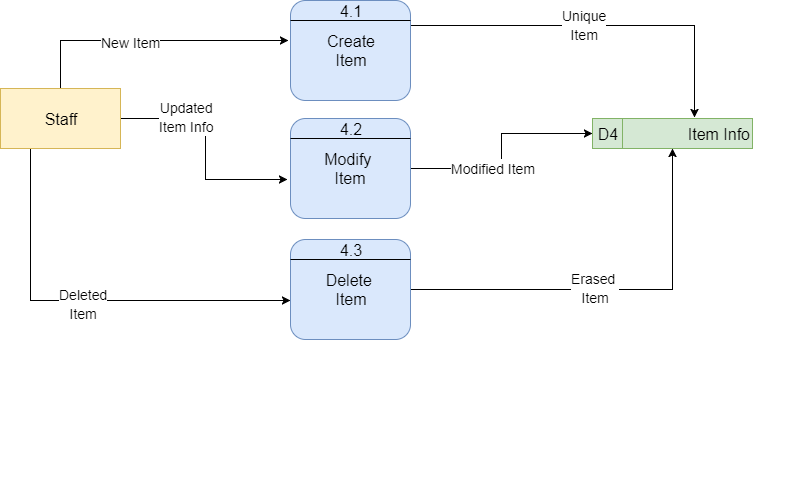
#### Manage Accounts



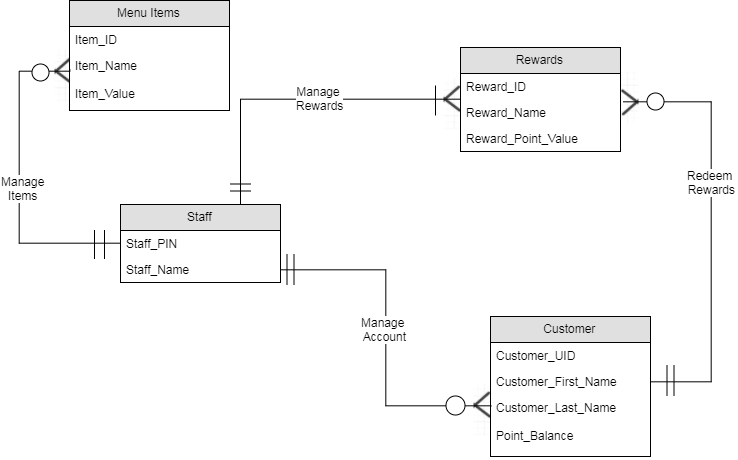
#### Manage Redeemable Rewards



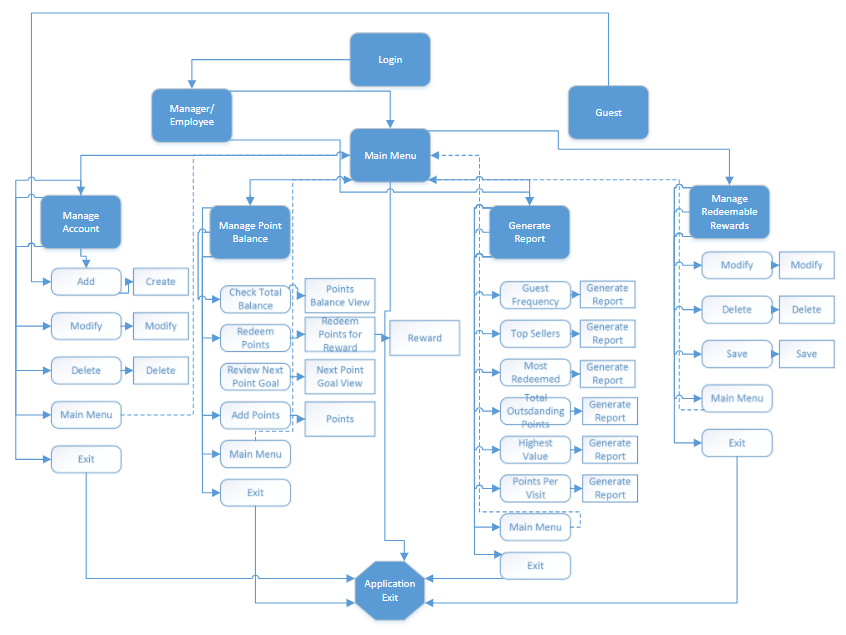
#### Manage Item Point Value



## Entity Relationship Diagram



## Navigation Diagram



## Program Plan

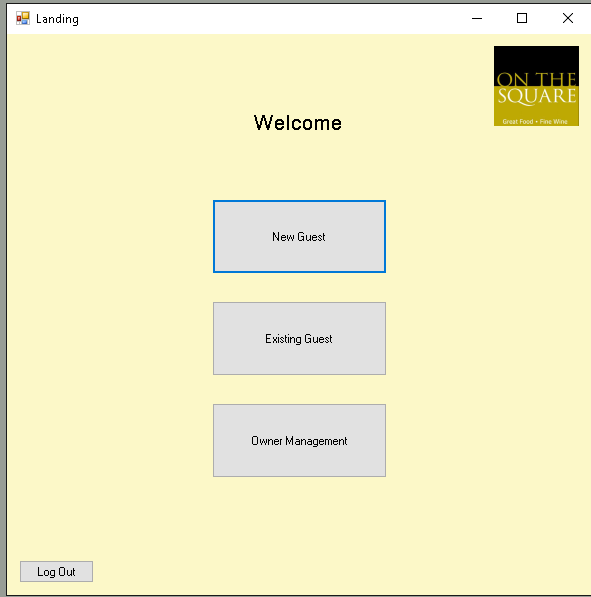
|  |  |  |  |
| --- | --- | --- | --- |
| **Form/Menu/Report** | **Input** | **Process** | **Output** |
| **Main Menu** | Manage Account Menu Select | Transfer to Manage Account menu | Load menu form |
|  | Login Menu Select | Transfer to Login Menu | Load menu form |
|  | Manage Point Balance Menu Select | Transfer to Manage Point Balance Menu | Load menu form |
|  | Generate Report Menu Select | Transfer to Generate Report Menu | Load menu form |
|  | Manage Redeemable Rewards Menu Select | Transfer to Manage Redeemable Rewards Menu | Load menu form |
|  | Manage Item Point Values Menu Select | Transfer to Manage Item Point Values Menu | Load menu form |
|  | Exit Select | Pop up message asking user "Are you sure you want to exit?” | Message Box |
|  |  | If user selects “Yes”, system closes | Application Closes |
|  |  | If user selects “No”, message box closes and stays on main menu form | Load main menu form |
| **Manage Account** | Form Load | Pull list of existing guest accounts from database and populate in Select Existing User drop box | Existing Guest Accounts List |
|  | Add Select | Test if user entered a new account name in “Create Guest Account” text box | Validation of text |
|  |  | If text contains value, pass to Add New Guest Form | Load Add New Guest Form and Pass New Guest Name and Phone Number Value |
|  |  | If text box does not contain a value, message user to enter a value | Message Box |
|  | Select Existing Guest drop box select | User selects from existing Guest Accounts | Guest Account Selection |
|  | Modify Select | Validate user has selected existing Guest Account | Validation of text |
|  |  | If selected, load Modify Guest Account form | Load Modify Guest Account Form |
|  |  | If no selection, message user to select existing Guest Account | Message Box |
|  | Delete Select | Validate user has selected existing Guest Account | Validation of text |
|  |  | If selected, load Delete Guest Account form | Load Delete Guest Account Form |
|  |  | If no selection, message user to select existing Guest Account | Message Box |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Create Guest Account** | Enter form values | Populate Guest Account from Manage Account menu | Guest First Name, Last Name, Phone Number, User ID |
|  |  | User fills in values | none |
|  | Create Select | Validate form entries for required fields and correct data types | Validated data |
|  |  | If any required values are missing or values contain incorrect data types, message user to resolve | Message Box |
|  |  | If all required values are populated and contain valid data types, add guest to database | New Guest Account |
|  | Manage Account Menu Select | Transfer to Manage Account Menu | Load Manage Account Menu |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Modify Existing Guest** | Form Load | Populate all values from database for selected guest account | Load data values for guest account |
|  | User Edits Values | System displays all altered values | New Values |
|  | Modify Select | Validate form entries for required fields and correct data types | Validated Data |
|  |  | If any required values are missing or values contain incorrect data types, message user to resolve | Message Box |
|  |  | If all required values are populated and contain valid data types, modify in database | Modified Guest Data |
|  | Manage Account Menu Select | Transfer to Manage Account Menu | Load Manage Account Menu |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Delete Existing Guest** | Form Load | Populate all values from database for selected guest account | Load data values for guest account |
|  | Delete Select | Message User “Are you sure you want to delete this guest?” | Message box |
|  |  | If user selects “Yes”, remove all guest data from database | Removed Data |
|  |  | If user selects “No”, return to form | Reload Form |
|  | Manage Account Menu Select | Transfer to Manage Account Menu | Load Manage Account Menu |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Login** | Employee enters five-digit pin | Validate correct pin number | Validation of Pin Number |
|  |  | If pin number is incorrect, system prompts user to “Please Try Again” | Message Box |
|  |  | If pin number is correct, system grants access to user | Load Main Menu |
| **Manage Point Balance** | Manage Point Balance Menu Select | Transfer to Manage Point Balance Menu | Load Manage Point Balance Menu |
|  | Check Total Balance Select | Transfer to Check Total Balance Form | Load Check Total Balance Form |
|  | Redeem Points Select | Transfer to Redeem Points Form | Load Redeem Points Form |
|  | Review Next Point Goal Select | Transfer to Review Next Point Goal Form | Load Next Point Goal Form |
|  | Add Points to Customer Balance Select | Transfer to Add Points to Customer Balance Select | Load Add Points to Customer Balance Form |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Check Total Balance** | Form Load | Populate all values from database for guest accounts | Load data values for guest accounts |
|  | Enter Guest Account User ID in search bar | Populate all values from database for Guest Account associated with User ID | Load data values for Total Points Balance |
|  |  | If User ID does not exist, system displays “There is no account associated with that User ID” | Message Box |
|  |  | If User ID exists, system displays guest information and total points balance | Load Guest data values and Total Points Balance |
|  | Manage Point Balance Select | Transfer to Manage Point Balance Menu | Load Manage Point Balance Menu |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Redeem Points** | Form Load | Populate all values from database for guest accounts | Load data values for guest accounts |
|  | Enter Guest Account User ID in search bar | Populate all values from database for Guest Account associated with User ID | Load data values for Total Points Balance |
|  |  | If User ID does not exist, system displays “There is no account associated with that User ID” | Message Box |
|  |  | If User ID exists, system displays guest information and total points balance | Load Guest data values and Total Points Balance |
|  | Redeem Points for Reward Select | Transfer to Rewards Selection page | Load Rewards Selection page |
|  | Select Reward | Subtract points associated with reward from guests account in database | Updated Points Balance |
|  |  | If not enough points in guest account to redeem selected reward, system displays “You do not have enough points to redeem this reward” | Message Box |
|  |  | If enough points are in the guest account to redeem selected reward, system displays “Your remaining points balance is \_\_\_\_” | Message Box |
|  | Manage Point Balance Select | Transfer to Manage Point Balance Menu | Load Manage Point Balance Menu |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Review Next Point Goal** | Form Load | Populate all values from database for guest accounts | Load data values for guest accounts |
|  | Enter Guest Account User ID in search bar | Populate all values from database for Guest Account associated with User ID | Load data values for current rewards and associated point values |
|  |  | If User ID does not exist, system displays “There is no account associated with that User ID” | Message Box |
|  |  | If User ID exists, system displays current rewards and associated point values | Load data values for current rewards and associated point values |
|  | Manage Point Balance Select | Transfer to Manage Point Balance Menu | Load Manage Point Balance Menu |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Add Points to Customer Balance** | Form Load | Populate all values from database for guest accounts | Load data values for guest accounts |
|  | Enter Guest Account User ID in search bar | Populate all values from database for Guest Account associated with User ID | Load data values for points available to add |
|  |  | If User ID does not exist, system displays “There is no account associated with that User ID” | Message Box |
|  |  | If User ID exists, system displays current rewards and associated point values | Load data values for current rewards and associated point values |
|  | Points select determined by price of meal | Populate points data value in database for associated User ID | Updated points value in database |
|  | Manage Point Balance Select | Transfer to Manage Point Balance Menu | Load Manage Point Balance Menu |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Generate Report** | Guest Frequency Select | Gather Frequency of Guest accounts from database | Generate Report |
|  | Top Sellers Select | Gather most selected item on menu from database | Generate Report |
|  | Most Redeemed Select | Gather most redeemed award from database | Generate Report |
|  | Total Outstanding Points Select | Gather Total Outstanding Points from database | Generate Report |
|  | Highest Value Members Select | Gather guest names and points from Highest to least from database | Generate Report |
|  | Points Per Visit Select | Gather guest names and points per visit from database | Generate Report |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |
| **Manage Redeemable Rewards and Point Values** | Reward Options Select | Populate data values with current rewards, associated points, and number of redemptions from database | Load Data Values |
|  | User Edits Name of Reward and Associated Points Values | System displays all altered values | New Values |
|  | Save Select | Update and save values in database | Saved and Updated Values |
|  |  | If duplicate item exists, system warns user “Item already exists” | Message Box |
|  | Main Menu Select | Transfer to Main Menu | Load Main Menu Form |

## Mock Forms

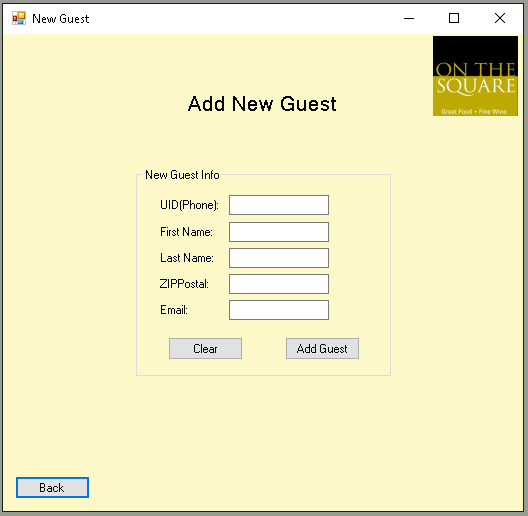
### Login



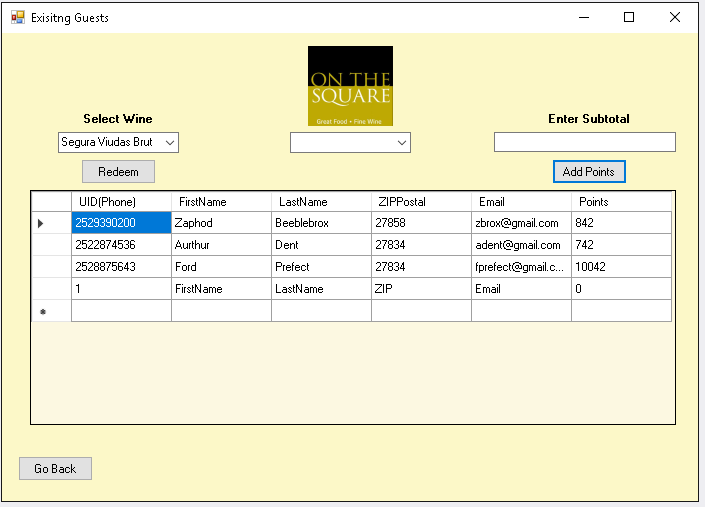
### Staff Menu

****

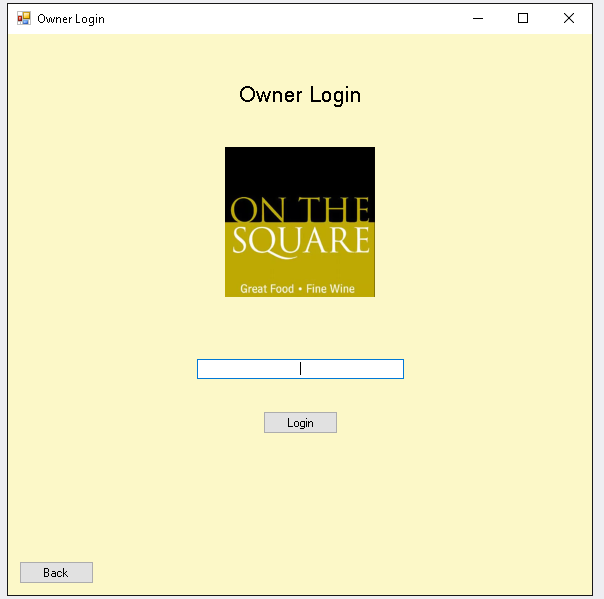
### Add New Guest



### Manage Guests/Rewards



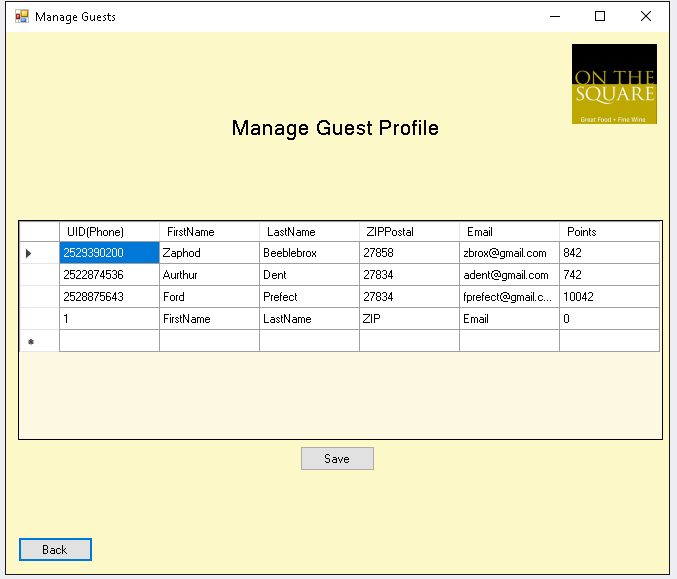
### Owner Login



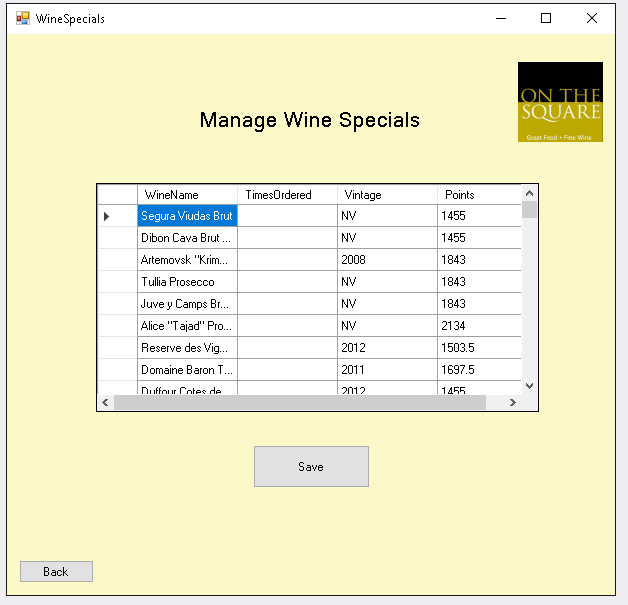
### Owner Menu



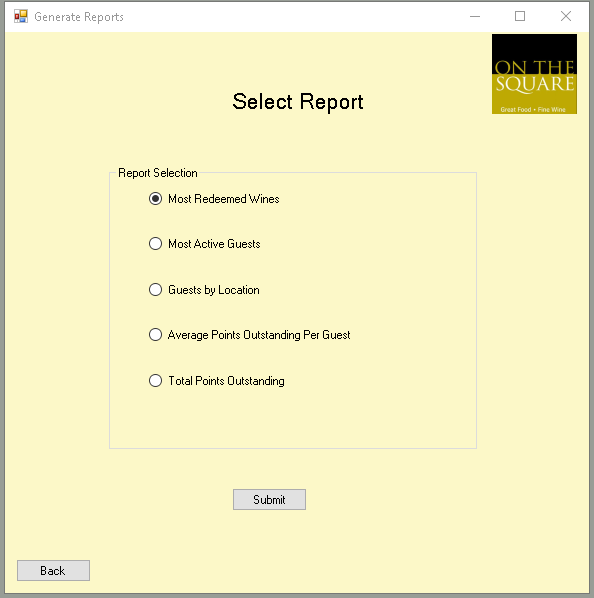
### Owner Guest Management



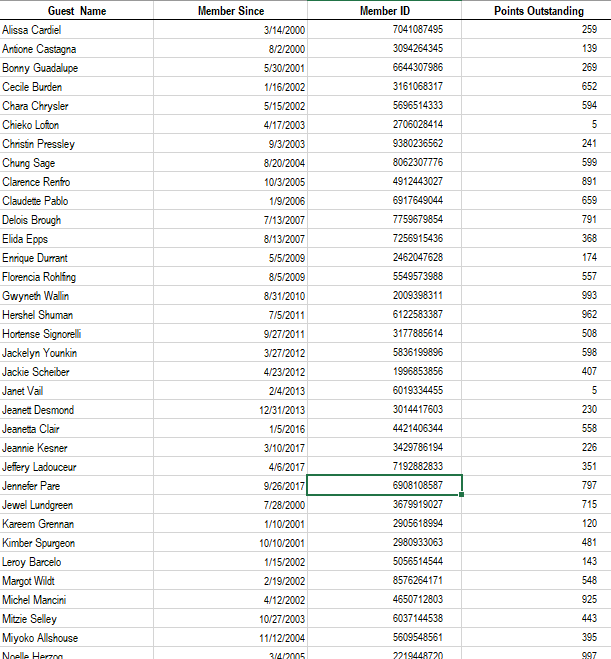
### Owner Wine Management

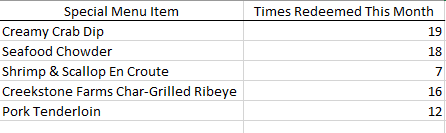


### Generate Reports



## Sample Reports







# Structured Walkthrough

## Challenges, Problems, and Discoveries – Test Plans

Some of the challenges we faced were working remotely and having conflicting schedules. Since most of our team lives hours away from the actual restaurant, it was hard to receive continuous feedback but we made it work in the end. Another challenge we faced was determining point values for each of the items. At first we were assigning points for each item ordered on the menu but realized that this would cause too much confusion and made it seem like we wanted customers to focus on earning more points instead of focusing on our previous goal which is customer retention. In the end, we decided to only assign points based on visits and the accumulated points will be used to redeem free glasses of wine.

## Program Code

### Login

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

}

}

}

### Landing

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class Landing : Form

{

public Landing()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

(new NewGuest()).Show();

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

(new Form2()).Show();

this.Close();

}

private void button3\_Click(object sender, EventArgs e)

{

(new OwnerLogin()).Show();

this.Close();

}

}

}

### New Guest

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class NewGuest : Form

{

[BrowsableAttribute(true)]

public AutoCompleteMode AutoCompleteMode { get; set; }

public NewGuest()

{

InitializeComponent();

}

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

{

this.Validate();

this.guestsBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.oTSDataSet);

}

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

{

this.guestsTableAdapter.Fill(this.oTSDataSet.Guests);

uID\_Phone\_TextBox.Text = String.Empty;

firstNameTextBox1.Text = String.Empty;

lastNameTextBox1.Text = String.Empty;

zIPPostalTextBox1.Text = String.Empty;

emailTextBox1.Text = String.Empty;

}

private void button1\_Click(object sender, EventArgs e)

{

this.Validate();

this.guestsBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.oTSDataSet);

guestsBindingSource.AddNew();

}

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

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

firstNameTextBox1.Text = String.Empty;

lastNameTextBox1.Text = String.Empty;

zIPPostalTextBox1.Text = String.Empty;

emailTextBox1.Text = String.Empty;

}

private void button2\_Click(object sender, EventArgs e)

{

(new Landing()).Show();

this.Close();

}

private void uID\_Phone\_TextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void firstNameTextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void lastNameTextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void zIPPostalTextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void emailTextBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void uID\_Phone\_TextBox\_TextChanged(object sender, EventArgs e)

{

}

}

}

### Add Guest Points

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class AddGuestPoints : Form

{

public AddGuestPoints()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

double Points,

Subtotal;

Subtotal = Convert.ToInt32(textBox1.Text);

Points = Subtotal \* .02;

MessageBox.Show(Convert.ToString(Points));

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) &&

(e.KeyChar != '.'))

{

e.Handled = true;

}

// only allow one decimal point

if ((e.KeyChar == '.') && ((sender as TextBox).Text.IndexOf('.') > -1))

{

e.Handled = true;

}

}

}

private void button2\_Click\_1(object sender, EventArgs e)

{

{

double Points,

Subtotal;

Subtotal = Convert.ToInt32(textBox1.Text);

Points = Subtotal \* .02;

MessageBox.Show(Convert.ToString(Points));

//\OTSDataSet obj = (OTSDataSet)

}

}

}

}

### Edit Existing Guests

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

this.wineTableAdapter.Fill(this.oTSDataSet.Wine);

this.guestsTableAdapter.Fill(this.oTSDataSet.Guests);

//

comboBox2.DisplayMember = "Text";

comboBox2.ValueMember = "Value";

List<DataRow> r = guestsTableAdapter.GetData().Select().ToList();

r.Reverse();

foreach (DataRow a in r)

{

string FullName = a["FirstName"] + " " + a["LastName"];

comboBox2.Items.Add(new { Text = FullName, Value = FullName});

}

comboBox2.SelectedIndex = 0;

}

private void button1\_Click(object sender, EventArgs e)

{

{

double Points, Subtotal;

if(!Double.TryParse(textBox1.Text, out Subtotal))

{

Console.WriteLine("Not a double");

// TODO add error messages

return;

}

Points = Subtotal \* .5;

var row = guestsDataGridView.CurrentRow;

if (row == null || row.Index < 0)

return;

var NewPoints = Convert.ToInt32(row.Cells["dataGridViewTextBoxColumn6"].Value) + Points;

guestsDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn6"].Value = +NewPoints;

this.Validate();

this.guestsBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.oTSDataSet);

guestsBindingSource.AddNew();

}

}

private void guestsDataGridView\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

this.Validate();

this.guestsBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.oTSDataSet);

}

private void button2\_Click(object sender, EventArgs e)

{

int WinePoints = 200;

var row = guestsDataGridView.CurrentRow;

if (row == null || row.Index < 0)

return;

var NewPoints = Convert.ToInt32(row.Cells["dataGridViewTextBoxColumn6"].Value) - WinePoints;

row.Cells["dataGridViewTextBoxColumn6"].Value = +NewPoints;

this.Validate();

this.guestsBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.oTSDataSet);

}

private void guestsDataGridView\_KeyPress(object sender, KeyPressEventArgs e)

{

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) &&

(e.KeyChar != '.'))

{

e.Handled = true;

}

// only allow one decimal point

if ((e.KeyChar == '.') && ((sender as TextBox).Text.IndexOf('.') > -1))

{

e.Handled = true;

}

}

}

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

{

}

private void comboBox1\_SelectedIndexChanged\_1(object sender, EventArgs e)

{

//wineDataGridView.SelectedRows.Clear();

foreach (DataGridViewRow row in wineDataGridView.Rows)

{

if (comboBox1.SelectedItem == wineDataGridView)

{

row.Selected = true;

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

(new Landing()).Show();

this.Hide();

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

foreach(DataGridViewRow a in guestsDataGridView.Rows)

{

a.Selected = false;

}

guestsDataGridView.Rows[comboBox2.SelectedIndex].Selected = true }

### Redeem Guest Points

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class RedeemGuestPoints : Form

{

public RedeemGuestPoints()

{

InitializeComponent();

}

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

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

}

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

{

this.Validate();

this.wineBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.oTSDataSet);

}

private void Form5\_Load(object sender, EventArgs e)

{

this.wineTableAdapter.Fill(this.oTSDataSet.Wine);

}

}

}

### Owner Login

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class OwnerLogin : Form

{

public string OwnerPIN = "1337";

public OwnerLogin()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

(new Form2()).Show();

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

if(textBox1.Text == OwnerPIN)

{

(new Landing()).Show();

this.Close();

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

}

}

### Owner Management

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class OwnerManagement : Form

{

public OwnerManagement()

{

InitializeComponent();

}

private void Form6\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

(new SelectReport()).Show();

this.Close();

}

}

}

### Manage Guest Profiles

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class ManageGuestProfiles : Form

{

public ManageGuestProfiles()

{

InitializeComponent();

}

}

}

### Select Report

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OTS\_Forms\_Revised

{

public partial class SelectReport : Form

{

public SelectReport()

{

InitializeComponent();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void Form7\_Load(object sender, EventArgs e)

{

guestsTableAdapter2.Fill(otsDataSet1.Guests);

wineTableAdapter2.Fill(otsDataSet1.Wine);

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

List<DataRow> rows = new List<DataRow>();

int Points = 0;

RadioButton checkedButton = groupBox1.Controls.OfType<RadioButton>().FirstOrDefault(r => r.Checked);

Form3 s = new Form3();

switch (checkedButton.Text)

{

case "Most Redeemed Wines":

rows = wineTableAdapter2.GetData().Select().OrderBy(u => u["TimesOrdered"]).ToList();

rows.Reverse();

foreach(DataRow r in rows)

{

s.addline(Convert.ToString(r["WineName"]) + ": " + Convert.ToString(r["TimesOrdered"]));

}

break;

case "Most Active Guests":

rows = guestsTableAdapter2.GetData().Select().OrderBy(u => u["Points"]).ToList();

rows.Reverse();

foreach (DataRow r in rows)

{

s.addline(Convert.ToString(r["Firstname"]) + " " + Convert.ToString(r["Lastname"]) + ": " + Convert.ToString(r["Points"]));

}

break;

case "Guests By Location":

break;

case "Average Points Outstanding Per Guest":

rows = guestsTableAdapter2.GetData().Select().ToList();

Points = 0;

foreach(DataRow r in rows)

{

Points += Convert.ToInt32(r["Points"]);

}

s.addline("Average Points Outstanding: " + Convert.ToString(Points / rows.Count));

break;

case "Total Points Outstanding":

rows = guestsTableAdapter2.GetData().Select().ToList();

Points = 0;

foreach (DataRow r in rows)

{

Points += Convert.ToInt32(r["Points"]);

}

s.addline("Total Points Outstanding: " + Convert.ToString(Points));

break;

default:

throw new Exception("Something went wrong here");

}

s.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

(new OwnerManagement()).Show();

this.Close();

}

}

}

## Test Data

The test data used to test the system was created with current vendor information and points where assigned based on the price of the wines. Fictitious guest information was used to test our database.

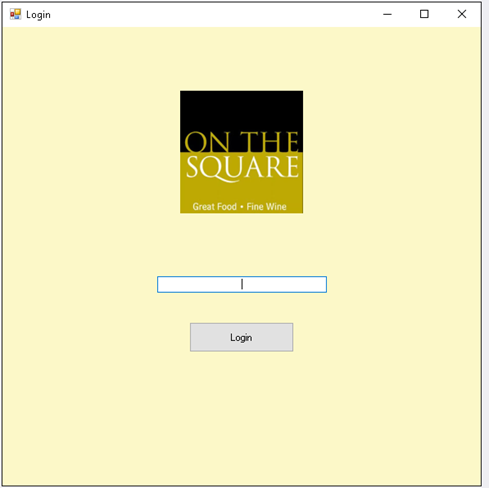
# Training

## [Staff Training Documentation](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561175)

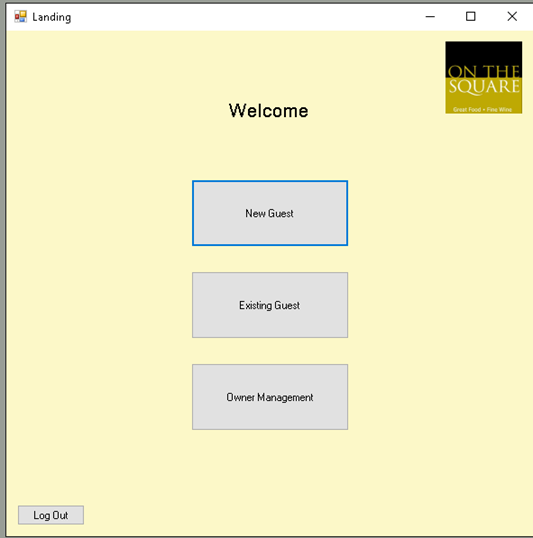
### [All Staff Functions:](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561176)

### [Login Screen](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561177)

* Here you will enter your PIN
* Once entered, press the Login button

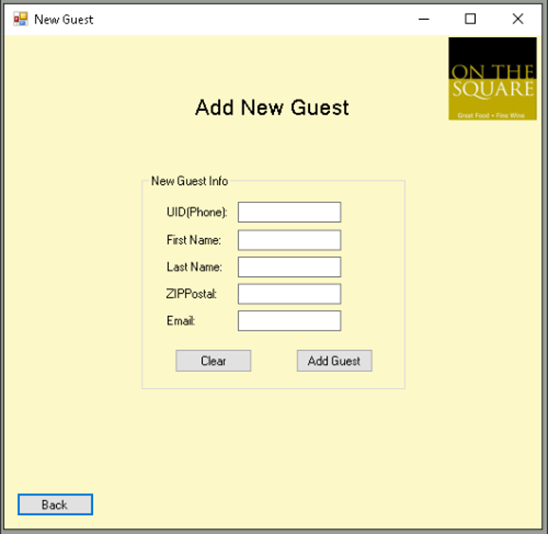
  
  
  
  
  
  
  
Landing screen

* The top button will take you to the New Guest window
* The middle button will take you to the Existing Guest window
* The bottom button will take you to the Owner Management window



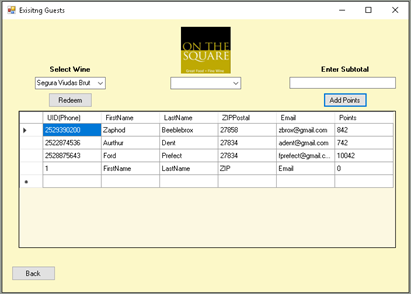
### [New Guest Information Window](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561178)

* Here you will enter in the New Guest information
  + The top text box allows input for the Guest’s phone number
  + The second text box allows input for the First Name
  + The third text box allows input for the Last Name
  + The fourth text box allows input for the Guest’s ZIP
  + The last text box allows input for the Guest’s Email
* Once all information is inputted, press the Add Guest button
* If you need to clear the entered text, you can press the Clear button to clear the data
* Once finished, you may press the Go Back button to return to the Landing screen



### [Existing Guests Window](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561179)

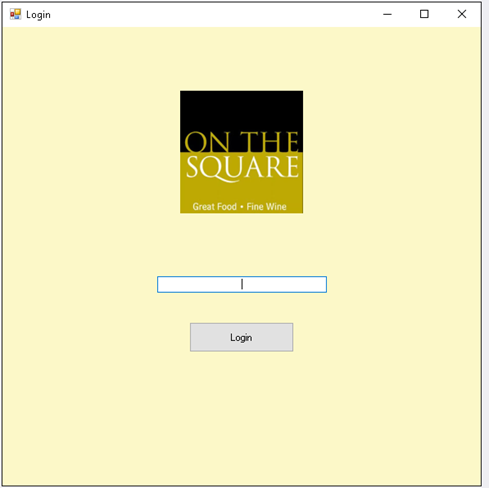
* The top right text box is where you will enter in the Guest’s subtotal from their receipt
  + Once the subtotal is inputted, press the Add Points button
* The top left text box is where you select the wine that the Guest would like to redeem
  + Once selected, press the Redeem button
* The middle combo box allows you to search for the Guest’s profile
* Once completed, press the Go Back button to return to the Landing screen



## [Owner Only Functions](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561180)

### [Owner Login](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561181)

* Here you will enter your PIN
* Once entered, press the Login button



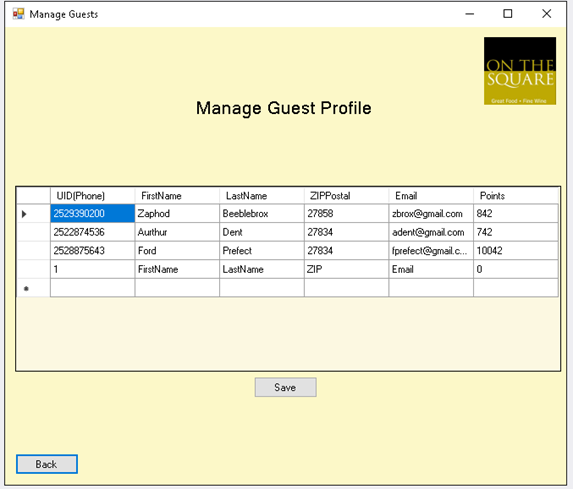
### [Owner Management Window](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561182)

* The top button will take you to the Manage Guests window
* The second button will take you to the Manage Wine Specials window
* The bottom button will take you to a Report Selection window to generate reports



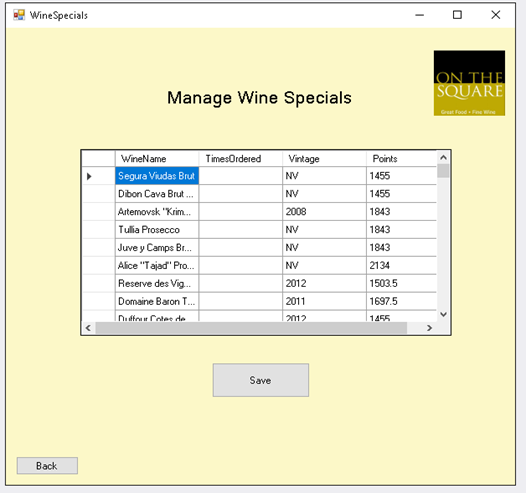
### [Manage Guest Window](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561183)

* This is the Manage Guests window
* Here you can select any cell to edit its values
* Once you are finished editing, press the Save button on the bottom of the screen
* Now you can press the Back button to return to the Owner Management screen



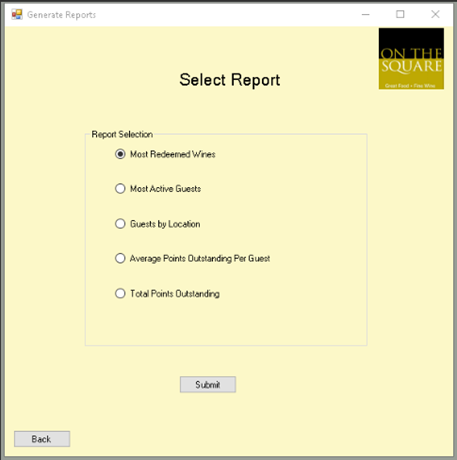
### [Manage Wine Specials Window](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561184)

* Here you can select any cell to edit specific cell values or add additional Wine Specials
* Once you are finished editing, press the Save button on the bottom of the screen
* Now you can press the Back button to return to the Owner Management screen



### [Report Selection Screen](file:///C:\Users\Paige\AppData\Local\Temp\Temp1_MS5.zip\MS5\Business_Buccaneers_Training%20Documentation.docx#_Toc498561185)

* You can select one report from this list at a time
* Once selected, you may press Submit
* You may close the Report window to return to this screen and select another Report to Generate
* Once finished, you may press the Back button to return to the Owner Management window



## Technical Manual

This technical documentation will assist you in ensuring a long prosperous life for your system. In this document we will discuss:

1. The database the system uses
2. Future growth for the system
3. Trouble Shooting
4. The different names and functions of the buttons

The program itself is a simple and straight forward design but we believe this should lead you to underestimate the value that’s within it. Instead that straightforward design will help you with preventing errors, and giving you room for future growth and strategy.

### Database

The Database has been written in Access, a Microsoft Office program. It contains two fields and 9 columns in total. This database is accessible through the OTSDataSet Access file. The fields, cells, values, columns, rows, and queries are all viewable and editable on the file. The only limit to the total amount of guests in the database, or wine you have saved is based on the storage space on your PC’s disk drive. But if you run this on the suggested machine with no other software installed or running you will have no problems at all running this for years and years, garnering an enormous database of loyal customers as time goes on.

#### The Guest Database

This is a locally saved access database field that contains the Guest’s information including their:

Phone: This is a string field that accepts both letters and numbers, this allows for prefixes if any need to be inserted for a phone number.  
FirstName: This is a string field that allows for up to 200 characters per entry, though you should reach that limit by entering most common names.   
LastName: This is a string field that allows for up to 200 characters per entry, though you should reach that limit by entering most common names.  
ZIPPostal: This is a string field that allows the Staff to enter in the Guests ZIP Code.  
Email: FirstName: This is a string field that allows for up to 200 characters per entry.  
Points: This is a Int field that only allows numbers which allows for quick calculations in our other functions.

#### The Wine Database

This is another locally saved Database, only this time it has only 3 columns:

WineName: A string field that allows 200 characters.  
Times Ordered: This is a int field that is increased everytime a wine is selected for redemption.  
Vintage: This is a short string field that allows for up to 4 characters, which gives the Owner enough room to put the year of the wine down in.

### Future Growth

This program has been written in a versatile adaptable format using C#. With the source code all the functions and methods can be edited by those with good familiarity with the C# language. If in the future you would like to add different functions to bring more value to the company, you can bring the provided source to a development company and express to them the feature you’d like and they should be able to add it to the program with very little difficulty.

Alternatively, you always have the option of contacting one of the team members who originally developed this software and ask them to add features, upon a fee to be agree on at the time depending on the function you’d like to add.

### Trouble Shooting

The program has been designed in a straight forward manner that shouldn’t experience any errors during normal functions. In the case of an error the first recommended quick fix is to simply close the software and reopen it. This will mean that you may need to reenter data that you were most recently putting in.

If you experience trouble with the database, you will need to open the Access Database and run trouble shooting there. A common one will be a duplicate entry, if this happens you can either edit the data in the Access Database or go to the manage guest function found in the Owner Management form.

If for whatever reason your PIN stops working you will need to contact one of our team members to check the error and source code to see what might be causing the PIN not to function. Though more commonly it will be a hardware problem relating to the input device.

### Forms

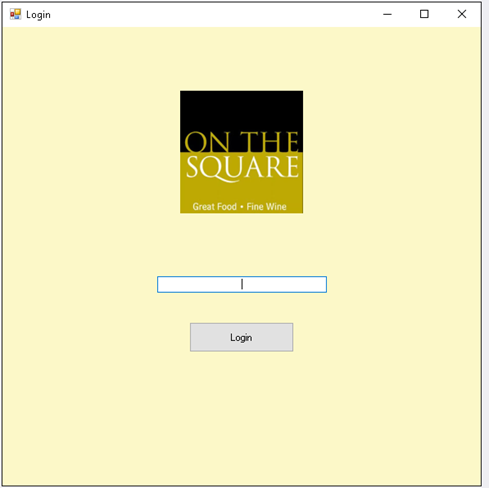
NOTE: All “clicks”, or “presses”, mentioned in this document should be performed by physically pressing the left button on your standard computer mouse.

#### Login Form:

The Login Form is where users log into the system, both the Owner and Employee PIN are accepted here, after they enter their correct PIN in the “Login Textbox” and press the “Login” button the window will change to “Landing” form.

This PIN is set in the code and is not adjustable outside of the source code. If at anytime you’d like to change the login PINs, add users, or remove PINs, it will need to be done inside the source code, which we’ve provided for you along with the rest of the program.

Window Functions:  
These three buttons are present in the entire program, so let’s get familiar with their function. The \_ button will minimize the window to your task bar to be pulled back up later. The ☐ button will maximize the window to take up all available screen space. The final X button closes, and quickly logs us out, of the program.

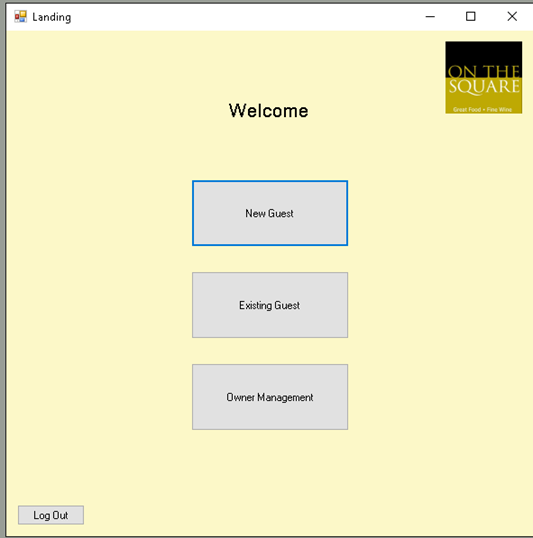


Login Textbox:  
This is where the owner and employees will enter in their PIN.

Login Button:  
After entering in their PIN the Owner or Employee will click this button to proceed into the Landing form, where most of our day to day functions will be.

#### Landing Form:

The Landing form is where most of the day to day features will be accessed. Here we have button access to the “New Guests” form, the “Existing Guest” form, and the “Owner Management” form. We will discuss these forms in more detail a little later. For not let’s just point out the “Log Out” button on the bottom left, it’s the only logout option that employees will have other than simply closing the window with the “X” button on the top right of the window we discussed in the “Login” form on the previous page.



Owner Management Button:  
By clicking this we can access the Owner Login form. This button should only be used by the Owner as it serves no functional purpose for Employees.

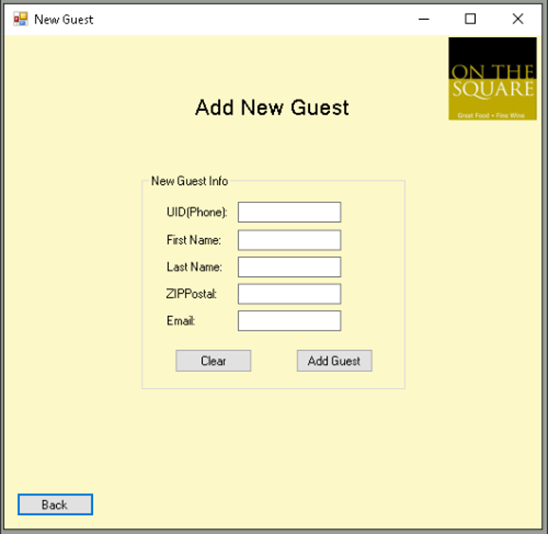
Existing Guest Button:  
By clicking this we can proceed to the Existing Guest Form.

New Guest Button:  
By clicking this we can proceed to the New Guest Form.

Lout Out Button:   
By clicking this button a user will be securely Logged out of the system, brining you back to the Login Form

#### New Guest Form:

Here on this form, accessed by clicking the New Guest Button, the owner and staff will have the ability to add new guests to the Guest Database, more information on this Database can be found on the Database page. The Owner and Employees will enter new guests on the occasion that guests would like to be added to the Loyalty program your restaurant offers. Without these guests this program is rendered useless, so be sure to collect as many guests as you can during business hours.

You will enter in standard Guest information including their:  
Phone Number  
First Name  
Last Name  
ZIP Code  
and their Email  
  


Back Button:  
Once a user has completed all you would like to on this form press this button to return to the Landing form

Clear Button:  
If at anytime a user would like to clear the above text boxes click this button and all above boxers will be cleared.

Add Guest Button:  
Once a user has entered all the guest’s information press this button to save it to the Guest Database.

Email Text Box  
Enter the guest’s Email here

ZIPPostal Text Box  
Enter the guest’s ZIP Code here

Last Name Text Box:  
Enter the guest’s last name here

First Name Text Box:  
Enter the guest’s first name here

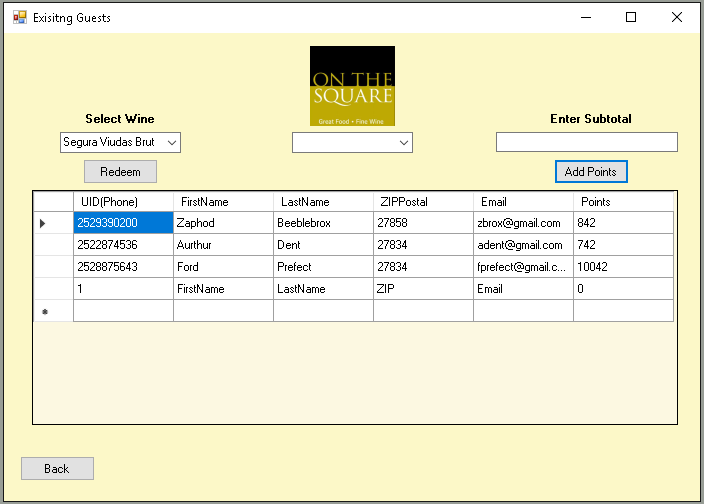
Phone Number Text Box:  
Enter the guest’s phone number here

#### Existing Guests Form:

This form is accessed by clicking the Existing Guest Button on the Landing form. Here is where Employees and the Owner will add and redeem points for the guest.

The functions here are what will increase your guest visits. The add points function will increase their total purchases, and the redeem will increase their interest in the wine catalog On The Square boats.

Below these functions the Owner and the Employees can view the Guest Database, but they cannot edit it, the database has been discussed in more detail above on the Database page.



Subtotal Text Box:  
In this box a user will enter in the selected guest’s subtotal. This subtotal will be found on the physical receipt.

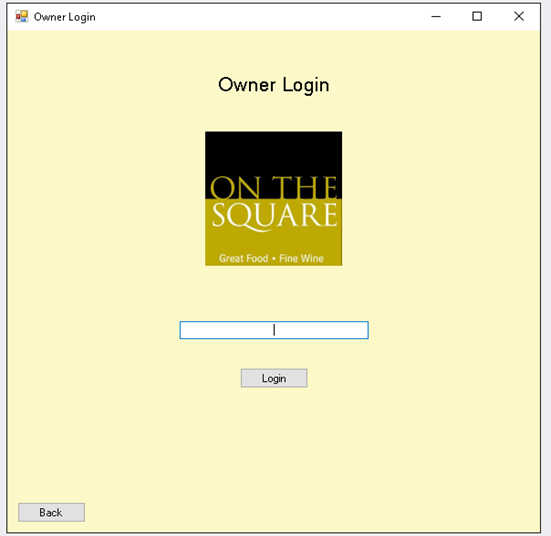
Select Wine Combo Box:  
By clicking this box a drop down will appear listing the wine available for redemption. Select the wine the guest wants to redeem.

Select Wine Combo Box:  
By clicking this box a drop down will appear listing the wine available for redemption. Select the wine the guest wants to redeem.

Add Points Button:  
After a user enters in the guest’s subtotal above, and they’ve selected their row in Guest Database, pressing this button will add points to their account in the points column.

Back Button:  
Once a user has completed all you would like to on this form press this button to return to the Landing form

#### Owner Login Form:

This form is accessed by clicking the Owner Management Button on the Landing form. The form here is very similar to the Login form, the only two differences here are the Back Button found on the bottom left and the fact that this Owner Login Text Box only accepts the Owner’s PIN.   
  


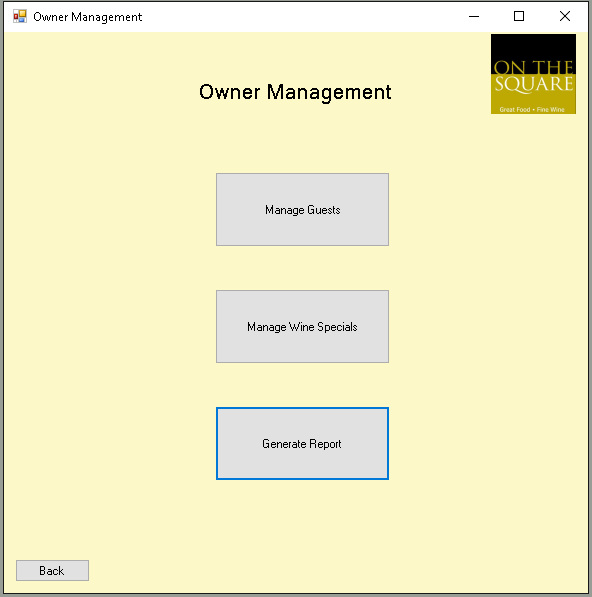
Login Textbox:  
This is where the Owner and will enter in their PIN.

Login Button:  
After entering in the Owner PIN the Owner or will click this button to proceed into the Owner Management form, where most of our management functions will be.

Back Button:  
If this form is accessed by an Employee on accident they may use this button to return to the Landing Form. Or an Owner may click this if they wish to return to the landing form as well.

#### Owner Management Form:

This form is only accessible to the Owner by entering his PIN on the previous Owner Login form. On this form we have access to the Manage Guests form, Manage Wine Specials form, and the Generate Reports Form. At the bottom left we have a Back Button that will take us to the Owner Login form.



Generate Report Button:  
By clicking this we can proceed to the Generate Report Form.

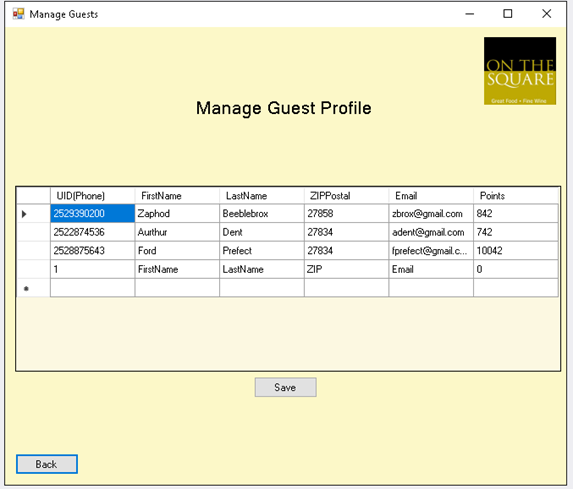
Manage Specials Wine Button:  
By clicking this we can proceed to the Manage Specials Form.

Manage Guest Button:  
By clicking this we can proceed to the Manage Guest Form.

Back Button:  
Once the Owner has completed all they would like to on this form press this button to return to the Owner Login form.

#### Manage Guest Profile:

This form is accessed by clicking the Manage Guests Button on the Owner Management form. This form gives the Owner full control over the columns and rows found in the local accessed Database. The Owner can click and cell below to edit it’s values to their liking. This may be done if the Owner gets latest information on a guest, or to correct a typo entered by any member of the Staff when they were entering a new guest on the New Guest form.

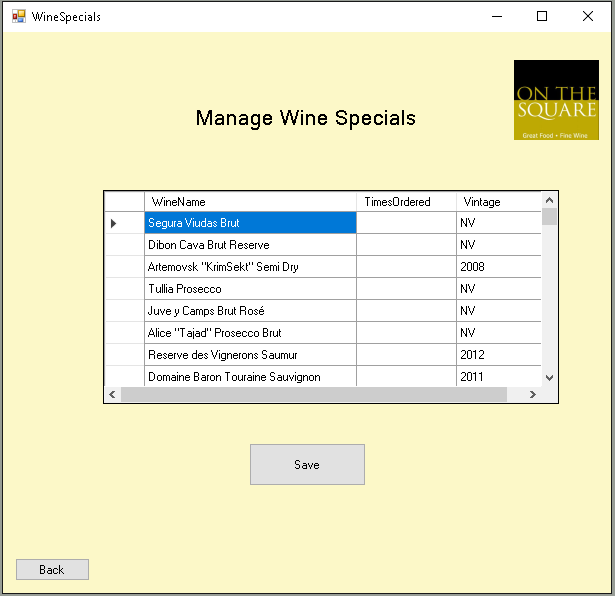


Guest Database:  
This is the Guest Database, here it and the individual cells are editable to any extent the Owner needs.

Back Button:  
Once a user has completed all you would like to on this form press this button to return to the Owner Management form

Save Button:  
Once the Owner is satisfied with their changes they may click this button to save the changed fields.

#### Manage Wine Specials form:

This form is accessed by clicking the Manage Wine Specials Button on the Owner Management form. This form will grant the Owner full modifiable access to the Wine Database. The Owner can click and cell below to edit it’s values if the Owner gets additional wines in stock and would like to add them to the redeemable wine list.  
  
  


Save Button:  
Once the Owner is satisfied with their changes they may click this button to save the changed fields.

Wine Database:  
This is the Wine Database, here it and the individual cells are editable to any extent the Owner needs.

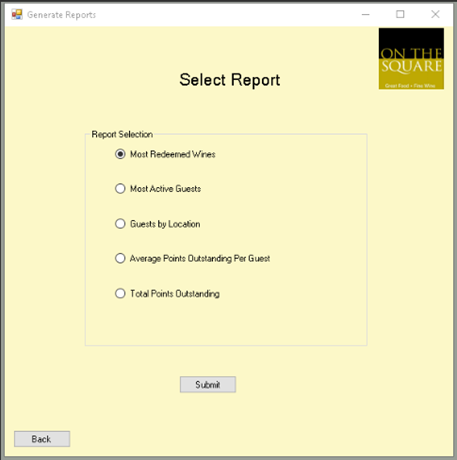
Back Button:  
Once a user has completed all you would like to on this form press this button to return to the Owner Management form

#### Select Report form:

This form is accessed by clicking the Generate Reports Button on the Owner Management form. The Owner will access this form when they want to generate reports to find our more information on their strategic move from the aspect of their guest behavior, their wine stock choices, and the performance and adoption of the point system.

There are 5 possible reports:

Most Redeemed Wines. This will give a top 10 list of the top most commonly redeemed wines, along with a number listing the total number of times each item has been ordered.  
Most Active Guests: This will give a list of the top 10 most active guests that utilize the points feature.  
Guest by Locations: This will give a top 10 list of the top most common ZIP codes, along with a number that sums the total instances the ZIP code is found in the Database  
Average Points Outstanding Per Guest: This report will take the total number of points outstanding and will divide it by the total guests in the system.  
Total Points Outstanding: This will give the Owner a report with a summation of the total number of points outstanding in the database.



Average Points Outstanding Per Guests Option:  
To view this report first click here.

Most Redeemed Wines Option:  
To view this report first click here.

Guests by Location Option:  
To view this report first click here.

Most Active Guests Option:  
To view this report first click here.

Most Redeemed Wines Option:  
To view this report first click here.

Back Button:  
Once a user has completed all you would like to on this form press this button to return to the Owner Management form

Report Submit Button:  
Once the Owner has selected their one desired report option, they will click here to generate the report.

# Project Assessment

## Johann Quintero (Business Analyst)

Throughout the course of this semester, I have found this capstone project to be very beneficial to me and my team. It gave us real world experience on how to design, build, and implement an information system for an actual company. Our system consisted of a customer relationship management (CRM) system to assist the restaurant On The Square with customer retention. The reason we chose to do this system was because we had identified On The Square had a problem with their customer retention. This is a very well known local restaurant in NC but they have a problem with customers not coming back.

My role as the Business Analyst was to identify what the problem was and how to fix it. As a team we decided we needed to implement some sort of rewards system where customers could accumulate points based on their visits and use those points to redeem different foods or desserts. Unfortunately, designing this system was not as easy as it seemed and we ran into a lot of roadblocks on the way. One of the challenges we faced was scope creep. Luckily, we caught it towards the first few milestones and made some major changes to our project. At first we were trying to assign points to different meals which made the menu seem like it was all about earning different types of points and wasn’t focused on customer retention. We realized this and fixed it to where points were accumulated based on amount spent and visits and could be redeemed for glasses of fine wine.

One of the main problems we had as a team was the distance and conflicting schedules. Each member of our group lives in different areas of NC and each have full time jobs or other commitments. This made scheduling meeting times a pain but in the end we would always find at least one day a week to meet. During our group meetings, each team member was assigned a task for our milestone and luckily we never ran into any problems with someone not completing their assigned tasks.

I feel that this course has been very helpful and has taught me a great amount of skills that will be needed in future projects at my jobs I may have. Our group worked extremely hard and are very passionate about our CRM system. One thing I realized is that if you do not have a passionate and committed team, then your project will most likely fail. I am glad I had such a hardworking group and we are very excited to see what this system will look like when it is actually implemented.

## Jack Fentzke (Database Coordinator)

Overall, our project has been successful with minimal downside. Although our team was stretched across the Nation most of the time, we were able to find ways to complete the tasks at hand and thus each milestone. The initial project company was not hard to come by, as one of our team members currently works there and has strong ties with the owner.

The initial idea was to create a program that would ultimately increase guest retention at On The Square, a restaurant in Tarboro, North Carolina. With over half the team having experience working with On The Square, it was a good start. Right away we started to determine how we would do this and thought of a Guest Point system, to encourage guests to come back and spend earned points.

It was not until the Milestone 3 Presentation that we realized we were a little off base as to how we would get guests to come back. At the time, our plan was to upsell menu items and push items that didn’t sell well to the guests with the reward of more points. This was not what we needed. This idea was pushing unwanted menu items to the guests to upsell and ultimately did not have the guest in mind; sure, we would add points to these items, but this was not the right way to achieve guest retention.

Immediately, we altered our plan to more reflect the true goal of guest retention. The team kept the idea of the point system; however, we made it much easier to earn the points. The guest would earn a flat rate of points just for visiting and additional points per the amount of money they spent while in the restaurant. The guest would then be able to spend these points on any of the Wine items that may be on the rewards list. The guest could get a glass of wine after their meal, or a bottle if they chose too. Points were also listed as possible payment for liquid desserts that On The Square offers. This method made earning points seem less like a hassle because they didn’t have to try new foods to earn points. The guest could come regularly, eat what they desire, and then have the option to possibly get a free glass of wine after dining in.

Once we figured out how we would increase guest retention we went to coding. The plan was to allow the Owner to oversee all aspects and then generate reports on the guest’s frequency, guest points outstanding, most redeemed wines, etc. We planned to use MS Access as our means of generating reports; however, once we began to code this work we realized it was a little outside of our knowledge. Ultimately, we created the databases within the application so the need for external databases was no more.

To conclude this assessment, I have benefited from this project and learned what it takes to do whatever is necessary. Poor leadership in the beginning of the semester should have been immediately taken care of, rather than wait halfway through the semester to take over and start to lead. I now know that if such ever occurs, it needs to be addressed immediately. Additionally, I learned how easy it is to get off track and miss your initial goal completely. Overall, I have had a pleasant experience working with the Team – Will Kozel, Johann Quinterro and Paige Lowery.

## William Kozel (Systems Analyst

This project has been actually quite stressful for me. Our team did a good job completing projects on time, but sometimes small details would get away from us until the day of submission and we would have to steamroll into it with full force in order to complete it on time. Though this happened several times, I think we did a decent job throughout the semester. Our group is composed of students who have recently taken on full time jobs, so while we’ve been working on this project, we’ve also been adapting to the career life, which, at least for me, has been a bit of a shock. Trying to find the balance between work and school and friends and family has been difficult, but we’ve managed.

Our project chose a great company to work with and they’ve been very supportive of us this entire time. I appreciate their interest in the project and their willingness to work with us. This attitude they’ve had has helped us create a program I think will benefit them greatly, though it may be sometimes they’re able to implement it as it’s currently very busy over there with the holiday season.

During this semester my role was technically Systems Analyst, but I, along with other groupmates, wore many hats in addition to this one. Each of us had a time working on various parts of the project we might not of expected to work on due to unforeseen problem areas, and unforeseen, but greatly appreciate, criticism from our group mates and professor.

Our team did a terrific job accomplishing the project change mid semester. We had to refocus on what was really important to the business, not what seemed to be a neat idea that has a proven track record at other businesses. Information systems are not always a cookie cutter problem solving project, though sometimes this can be the case. The mid semester project direction switch did present some problems, but our team was willing to work with them and hit them head on, adapting quickly, thanks to the agile methodology.

I appreciate the work I had to do, and the course’s challenging workload. It tested my ability and what I was capable of. This course makes me look forward to working in a MIS field job, and has helped me build up the courage to quit my current job at the end of this semester.

## Paige Lowery (Project Manager)

Overall, this project has been a really great experience for my team as well as myself. This project has always been a passion of mine, due to the fact that I was an employee at On The Square for 4 years and hold the restaurant, owners, and staff all close to my heart. When I heard that my team and professor were going to approve of my idea of working with the company to create a system that would help them grow and become more modern I was so excited.

My role as the Project Manager was to lead the team to complete all our deliverables on time, make sure we stayed on task, delegate work, and to meet with the owners of the restaurant to ensure that their needs are being met as well.

With the combined knowledge of each of my team members we could complete a successful program that worked for the needs of the restaurant. Will was our programmer who did most of the coding, Jack designed a lot of the forms with Will and created the database’s, and Johann created all the navigation diagrams and specifications for the program. There was much more work that went into each member’s contribution, but these are just a few to highlight how each of us could work together.

In the end, my team members are people that I can rely on professionally and on a personal level and would trust them to help me accomplish more tasks like this project. I am also proud of the project and feel confident taking it back to the owners of the restaurant to use and maintain in their business.

## Project Plan

|  |  |  |
| --- | --- | --- |
| **Component** | **Assigned To** | **Due Date** |
| **MS 1 - Conceptual Walkthrough** | Group | **9/13** |
| Company Background | Group | 8/27 |
| Current Environment | Will | 8/30 |
| Reason for Selection | Will | 8/30 |
| Discussion of Problem | Will | 8/30 |
| System Obj. and Constraints | Jack | 9/2 |
| System Requirements | Johann | 9/7 |
| Expected Benefits | Paige | 9/7 |
| Stakeholders | Johann | 9/10 |
| Scope Discussion | Paige | 9/10 |
| Problem Analysis | Group | 9/10 |
| **MS 2 - Analysis Presentation** | Group | **9/27** |
| Additional Suggestions/Improvements/Changes | Group | 9/14 |
| Context Diagram (NOT THE DFD) | Jack | 9/25 |
| Use Cases | Jack, Johann, Will | 9/23 |
| Data Flow Diagram | Johann | 9/26 |
| Appropriate Elicitation to Gather Information | Will | 9/20 |
| Develop Requirement Documentation | Paige | 9/22 |
| Non-Functional Requirements Artifacts | Johann | 9/21 |
|  |  |  |
| Documentation Deliverables | Paige | 9/25 |
|  |  |  |
|  |  |  |
| **MS 3 - Design Presentation** | Group | **10/18** |
| Exec. Summary w/ Narrative & Conclusions | Group | 10/17 |
| Complete DFD | Paige | 10/12 |
| Detailed Non-Functional Requirements | Will | 10/15 |
| Program Plan (With Logic & I/O's) | Johann | 10/14 |
| Hardware and Software Specifications | Will | 10/15 |
| Navigation Diagram | Johann | 10/14 |
| Mockup Forms and Reports | Jack | 10/15 |
| Sample Reports w/ Data | Jack and Will | 10/16 |
| Entity Relationship Diagram | Paige | 10/15 |
| *Include Proper/Standard Naming Conventions* | Jack | 10/15 |
| **MS 4 - Structured Walkthrough** | Group | **11/1** |
| Database Development | Will | 17-Oct |
| Program Development | Jack | 20-Oct |
| Test Plan | Johann | 25-Oct |
| Updated Project Plan and Timesheets | Paige | 29-Oct |
| Milestone Evaluation | Group | 1-Nov |
| **MS 5 - Training Presentation** | Group | **11/15** |
| Choice of Training Mode | Group | 10-Nov |
| Narrative and Screen Shots from Presentation | Will | 13-Nov |
| Documentation | Paige | 13-Nov |
| Template for Training New Users | Jack and Johann | 13-Nov |
| **MS 6 - Final Presentation** | Group | **12/6** |
| Updated Project Plan and Timesheets | Paige | 12/1 |
| Complete Training with users | Group | 12/2 |
| Final Presentation | Group | 12/3 |
| Group Member's Evaluations of the Project | Group | 12/4 |
| Final Testing of Program (Debugging) | Will | 12/5 |
| Updating Final Deliverables | Group | 12/6 |

# Appendix

## Accomplishments

Throughout the entire semester, our team proved to be hard workers and able to come together to collaborate on a project to create a successful running system for our company. Each milestone that we encountered we delegated into different parts for each team member and every single time each person could deliver what was asked of them.

## Problems Encountered

Taking this class online and not being able to meet with our team members face-to-face for each milestone was a challenge, but we overcame it by using our experience in communicating virtually and made it work. Another issue we faced was that the company we worked with had such an outdated system that we had to construct something completely new and train all the users. In the middle of the semester, we got some feedback that made us realize we were too focused on the points and not focused on the fine dining atmosphere or the wine. So we took into account the recommendations and made some major changes to the system which allowed us to create a better program for the restaurant. The timeline was our last issue, we could have always used more time to make those last-minute touches we wanted, but at the end our project ended up turning out the way we wanted.

## Timesheets

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MS 1** |  | **Task** |  | **1-Sep** | **8-Sep** | **11-Sep** | **Invidual Total** | |
|  |  |  |  |  |  |  |  |  |
| **Will Kozel** | | Current environment, project background, problems with current system |  | 2 hours | 2 hours | 2 hours | 6 hours |  |
| **Johann Quintero** | | System requirements, users involved in or affected by project |  | 2 hours | 2 hours | 2 hours | 6 hours |  |
| **Paige Lowery** | | Company Background and Description, Expected benefits of proposed system, Scope |  | 2 hours | 2 hours | 2 hours | 6 hours |  |
| **Jack Fentzke** | | System objectives, features, and constraints |  | 2 hours | 2 hours | 2 hours | 6 hours |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  | **Total Per Meeting** | 8 hours | 8 hours | 8 hours |  |  |
|  |  |  | **Overall** | 24 hours |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **MS 2** |  | **Task** |  | **14-Sep** | **20-Sep** | **23-Sep** | **Invidual Total** | |
|  |  |  |  |  |  |  |  |  |
| **Will Kozel** | | Proposed system objectives and restraints, fully detailed use case specifications |  | 1 hour | 2 hours | 3 hours | 6 hours |  |
| **Johann Quintero** | | Stakeholders, use case executive summary, non-functional requirements |  | 2 hours | 1 hour | 2 hours | 5 hours |  |
| **Paige Lowery** | | Executive background, current business enviroment, Expected benefits, organize Milestone 2 documents and deliverables |  | 1 hour | 1 hour | 3 hours | 5 hours |  |
| **Jack Fentzke** | | Context diagram, use case diagram, |  | 2 hours | 1 hour | 3 hours | 6 hours |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  | **Total Per Meeting** | 6 hours | 5 hours | 11 hours |  |  |
|  |  |  | **Overall** | 22 hours |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **MS 3** |  | **Task** |  | **14-Oct** | **15-Oct** | **16-Oct** | **Invidual Total** | |
| **Group** |  | Exec. Summary w/ Narrative & Conclusions |  | 1 hour |  |  |  |  |
| **Paige** |  | Complete DFD |  |  | 4 hours |  |  |  |
| **Will** |  | Detailed Non-Functional Requirements |  |  | 2 hours |  |  |  |
| **Johann** |  | Program Plan (With Logic & I/O's) |  | 2 hours |  |  |  |  |
| **Will** |  | Hardware and Software Specifications |  |  |  | 3 hours |  |  |
| **Johann** |  | Navigation Diagram |  |  | 3 hours |  |  |  |
| **Jack** |  | Mockup Forms and Reports |  | 4 hours |  |  |  |  |
| **Jack and Will** | | Sample Reports w/ Data |  |  | 2 hours |  |  |  |
| **Paige** |  | Entity Relationship Diagram |  |  |  | 2 hours |  |  |
| **Jack** |  | *Include Proper/Standard Naming Conventions* |  |  |  |  |  |  |
|  |  |  | **Total Per Meeting** | 7 hours | 11 hours | 5 hours |  |  |
|  |  |  | **Overall** | 23 hours |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **MS 4** |  | **Task** |  | **17-Oct** | **20-Oct** | **29-Oct** | **Individual Total** | | |
| **Will** |  | Database Development |  | 2 hours | 2 hours |  | 4 hours |  |
| **Jack** |  | Program Development |  | 2 hours | 1 hour | 2 hours | 5 hours |  |
| **Johann** |  | Test Plan |  |  | 2 hours |  | 2 hours |  |
| **Paige** |  | Updated Project Plan |  | 1 hour |  |  | 1 hour |  |
| **Group** |  | Milestone Evaluation |  |  |  | 2 hours | 2 hours |  |
|  |  |  | **Total Per Meeting** | 5 hours | 5 hours | 4 hours |  |  |
|  |  |  | **Overall** | 14 hours |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **MS - 5** |  | **Task** |  |  |  |  | **Individual Total** | |
| **Group** |  | Choice of Training Mode |  | 1 hour |  |  | 1 hour |  |
| **Will** |  | Narrative and Screen shots |  |  | 2 hours | 2 hours | 4 hours |  |
| **Paige** |  | Documentation |  |  |  | 3 hours | 3 hours |  |
| **Jack and Johann** | | Template for Training New Users |  |  | 2 hours | 2 hours | 4 hours |  |
|  |  |  | **Total Per Meeting** | 1 hour | 4 hours | 7 hours |  |  |
|  |  |  | **Overall** | 12 hours |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| **MS 6** |  | **Task** |  | **2-Dec** | **3-Dec** | **6-Dec** | **Individual Total** | |
| **Paige** |  | Updated Project Plan and Timesheets |  | 1 hour | 2 hours |  | 3 hours |  |
| **Group** |  | Complete Training with users |  | 2 hours |  | 2 hours | 4 hours |  |
| **Group** |  | Final Presentation |  | 1 hour | 1 hour | 1 hour | 3 hours |  |
| **Group** |  | Group Member's Evaluation of the Project |  | 1 hour | 1 hour | 1 hour | 3 hours |  |
| **Will** |  | Final Testing of the Program (Debugging) |  | 1 hour |  | 2 hours | 3 hours |  |
| **Group** |  | Update Final Deliverables |  | 1 hour | 1 hour | 3 hours | 5 hours |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  | **Total Per Meeting** | 7 hours | 5 hours | 9 hours |  |  |
|  |  |  | **Overall** | 21 hours |  |  |  |  |