

# System Design Project 2018

## Last Resort

## Table of Contents

Executive Summary.....	1
Technology Requirements .....	2
Expected Timeline.....	2
Actor List and User Stories.....	3
Actor List .....	3
Generic.....	3
Specific Role .....	3
Specific Department .....	3
User Stories.....	4
Use Case Narratives .....	13
Food Operations .....	13
1.1.05 Create Recipes .....	13
1.1.09 Add Recipes.....	14
1.1.11 Provide information about the restaurant .....	15
1.1.16 Create Food Ticket .....	16
1.2.14 Browse recipes.....	17
1.2.21 View Sold Tickets .....	18
1.2.24 Search Recipes .....	19
1.2.28 Browse Orders .....	20
1.2.29 View Recipes Details .....	21
1.3.07 Add Food Item.....	22
1.3.10 Update Recipes .....	24
1.4.05 Delete Recipes.....	25
1.4.08 Remove Dishes from Menu.....	26
2.1.02 Add Drink .....	27
2.1.17 Add Menu List .....	28
2.2.01 Browse Menu List.....	29
2.2.05 View Recipe Details.....	30
2.2.06 View Orders.....	31
2.3.11 Edit Drink.....	32
2.4.05 Remove Recipes.....	33
2.4.08 Deactivate Menu List .....	35

3.1.05 Create Catering Request .....	36
3.3.13 Update Catering Request.....	37
Supply Operations.....	38
1.1.08 Create Supply .....	38
1.2.18 Browse Inventory.....	39
1.3.13 Update Quantities to Correct Mistakes .....	40
1.2.17 Monitor Demand and Document Inventory .....	41
1.4.07 Cancel Orders.....	42
2.1.14 Browse Inventory System .....	43
2.2.24 Search Order Supplier .....	44
2.2.51 Browse Order Supply .....	45
2.2.54 Search for Supplier Account.....	46
2.3.22 Edit Order Supplies .....	47
2.4.14 Cancel Order Supplies.....	48
3.1.16 Create New Supplier .....	49
3.2.25 Browse Suppliers.....	50
3.3.19 Update Supplier .....	51
4.2.17 Search Supplier Account .....	52
4.2.37 Access Supply Request.....	53
5.2.10 View Supply Order Details .....	54
Facility Operations .....	55
1.2.36 Browse Hours.....	55
2.1.16 Create Maintenance Ticket.....	56
2.2.34 View Maintenance Work Orders .....	57
3.2.32 Read Building List .....	58
5.1.02 Add Reservation.....	59
5.2.01 View Details of Reservation .....	60
5.2.06 View List of Reservations .....	61
5.2.08 View Maintenance Ticket Details.....	62
5.2.09 View Room Details .....	63
5.3.15 Update Maintenance Ticket.....	64
5.4.03 Deactivate Open Maintenance Slip.....	65
5.4.06 Deactivate Room.....	66

Guest Operations .....	67
1.3.20 Checkout .....	67
3.4.11 Delete Account.....	68
5.5.01 Guest Check In .....	69
5.3.06 Edit Room Reservation.....	70
5.4.02 Remove Reservation .....	71
6.1.06 Create User Account .....	72
6.4.04 Deactivate Guest Account.....	74
Personnel Management.....	75
1.2.01 Log In.....	75
1.2.02 Log Out.....	76
1.1.19 Create New Employee Role .....	77
1.1.20 Create New Employee Schedule .....	78
1.2.45 Retrieve Employee Roles .....	79
1.2.49 Retrieve Employee Schedule .....	80
1.2.50 Retrieve Browse Employee Schedules.....	81
1.3.23 Update Employee Roles .....	82
1.3.24 Update Employee Schedule .....	83
1.4.13 Delete Employee Roles .....	84
1.4.15 Delete Schedules.....	85
2.2.39 View Total Daily Tips .....	86
3.2.21 Marketing View Budget .....	87
3.2.48 Browse Account .....	88
3.3.29 Update Account .....	89
6.1.08 Create a Message.....	90
Event Operations .....	91
2.2.50 Search for a Schedule Event.....	91
3.1.03 Create Event.....	92
3.1.04 Create Event Request.....	94
3.1.06 Create Advertising Event.....	95
3.1.08 Create New Sponsor .....	96
3.1.10 Create Account.....	97
3.2.23 Read Advertising Event .....	98

3.2.31 Read Event Request .....	99
3.2.35 Read Event .....	100
3.2.36 Read Setup List.....	100
3.2.37 Read Performance Details .....	102
3.2.38 Read Account Details .....	103
3.2.44 Browse Event Request .....	104
3.2.45 Read All Events.....	104
3.2.46 Browse Performance.....	105
3.2.49 Browse Advertising Event .....	106
3.2.52 Browse Setup List.....	107
3.3.15 Update Advertising Event .....	108
3.3.26 Update Performance.....	109
3.3.30 Update Event Request .....	110
3.4.10 Delete Scheduled Performance .....	111
3.4.12 Delete Event Request.....	112
3.4.14 Delete Events .....	112
<b>Service and Appointment .....</b>	<b>114</b>
3.2.30 Read Resort Services.....	114
3.2.24 Read Review.....	115
3.2.47 Browse Resort Service.....	116
3.2.53 Browse Reviews .....	117
5.2.27 Retrieve Vehicle by License Plate .....	117
5.2.28 Retrieve Vehicle by Guest Name .....	118
5.2.37 Retrieve Shuttle Vehicle by Vin.....	119
5.2.38 Retrieve Shuttle Vehicle by License Plate .....	120
5.3.12 Update Vehicle's Parking Spot .....	121
5.3.16 Update Shuttle Vehicle Status .....	122
5.4.05 Deactivate Shuttle Vehicle .....	123
6.2.05 View List of Children .....	124
6.2.06 View Child Details .....	125
6.2.23 Update Dependent Record .....	126
7.1.01 Add a New Dependent .....	127
7.1.03 Sign Pets Up for Grooming Appointments.....	128

7.2.14 Groomer Appointment Book .....	130
7.2.16 Show Appointment Details .....	131
7.2.21 Browse Pets.....	131
7.3.14 Change an Appointment Details .....	132
7.4.05 Delete Appointment .....	134
Product Operations.....	135
1.2.39 View Products .....	135
1.2.40 View Cart.....	136
2.1.21 Create Report.....	137
2.2.22 Search Item .....	139
2.4.13 Delete Item .....	140
3.2.50 Browse Package .....	141
3.4.06 Delete Package.....	142
4.1.04 Create Product .....	143
4.2.04 View Product.....	144
4.2.41 Browse Items .....	145
4.3.05 Edit Product.....	146
4.4.02 Deactivate Product.....	147
Use Case Diagrams.....	150
Food Operations .....	150
Supply Operations.....	151
Facility Operations .....	152
Guest Operations .....	154
Personnel Management.....	157
Event Operations .....	159
Service and Appointment .....	165
Product Operations.....	170
Activity Diagrams .....	172
Customer Operations.....	172
Catering Request.....	172
Request Event .....	173
Reserve Appointment .....	174
Reserve Resort Room.....	175

Reserve Spot in Number Dependent Event .....	176
Process Food Order.....	177
Food Request .....	178
Materials .....	179
Confirm Order.....	179
Delivery .....	180
Personnel Management.....	181
Appointment.....	182
Purchases and Checkout.....	183
Event Schedule.....	184
Maintenance Work Order Viewer.....	185
Maintenance Ticket .....	187
Manager – Packages .....	188
Manager – Supplies.....	189
Performance Schedule.....	190
Employee Time-Off Request .....	191
Work Schedule .....	192
Interfaces .....	193
Internal.....	193
Food Operations .....	193
Supply Operations.....	196
Facility Operations .....	203
Guest Operations .....	218
Personnel Management.....	233
Event Operations .....	251
External .....	274
Supply Operations.....	275
Guest Operations .....	279
Event Operations .....	287
Sequence Diagrams.....	297
Automation .....	317
Work Scheduling .....	317
Class Diagram.....	317

State Diagram.....	318
Sequence Diagram .....	319
Ordering Supplies.....	319
Class Diagram.....	320
State Diagram.....	321
Sequence Diagram .....	322
Reservations.....	322
Class Diagram.....	323
State Diagram.....	324
Sequence Diagram .....	324
Event Builder.....	325
Class Diagram.....	325
State Diagram.....	326
Sequence Diagram .....	327
Class Diagram.....	328
Data Layer .....	348
ER Diagram.....	348
Data Dictionary .....	348
EMPLOYEE .....	348
ROLE .....	349
EMPLOYEE ROLE.....	349
DEPARTMENT.....	349
MEMBER .....	349
GUEST.....	350
GUEST TYPE.....	350
RESERVATION.....	350
MEMBER TAB .....	351
MEMBER TAB LINE .....	351
GUEST VEHICLE .....	351
ROOM.....	351
ROOM RESERVATION.....	352
ROOM LINE.....	352
ROOM TYPE.....	352

BUILDING.....	352
MAINTENANCE WORK ORDER .....	353
MAINTENANCE TYPE .....	353
SERVICE COMPONENTS.....	353
EVENT.....	354
EVENT TYPE .....	354
EVENT REQUEST.....	354
SETUP LISTS.....	355
SETUP LIST LINE.....	355
SETUP LIST TASKS.....	355
PERFORMANCE .....	355
EVENT PERFORMANCE.....	356
OFFERING.....	356
OFFERING TYPE .....	356
RECIPE .....	356
RECIPE ITEM LINE.....	357
ITEM .....	357
ITEM TYPE .....	357
SUPPLIER .....	357
ITEM SUPPLIER.....	358
ITEM ORDER.....	358
ITEM ORDER LINE.....	358
PET.....	359
RESORT VEHICLE .....	359
VEHICLE CHECKOUT .....	360
SCHEDULED ITEM.....	360
APPOINTMENT.....	360
APPOINTMENT TYPE .....	360
SHOP .....	361
PET TYPE.....	361
EVENT EMPLOYEE .....	361
SUPPLY ORDER .....	361
SUPPLY ORDER LINE .....	362

Stored Procedures .....	362
Contributors.....	371

## Executive Summary

Millennial Resorts is a full service resort that focuses on giving their guests a great experience during their stays. Last Resort Software Company, Inc. (LRSC, Inc.) feels that the resort experience can be enhanced by upgrading its operation and information systems software with our application. LRSC believes this would make it easier for Millennial associates to serve their guests. Modern streamlined systems will put the focus of resort associates where it should be, on the guests, and not the everyday tasks of running the resort. In addition, guests need the ability to interact with the resort's services and should be able to do such things as sign up for scheduled events and make appointments. By giving the guest more control and information their resort experience will be more enjoyable.

The documentation explains the Event Builder; a way for an event planner to build all the necessary elements of an event in one location. It will let the planner know at each step of the process if the resources are available to accommodate the event. This will streamline event planning. These efficiencies will encourage use of more of what the resort has to offer. The activity diagrams detailed below show the various guest interactions and their progression within the resort such as purchasing goods and services, reserving rooms, and utilizing amenities. Some activity diagrams detail the progressive steps of managing the resort inventory, processing orders, creating and managing items, and vendor management.

Millennia Resorts Software Application is a community blend of locally sourced activities for guests, kids, and pets. The User Stories documentation shows a detailed look at the gathered activities that each user needs to accomplish and what they need the software to do to accomplish that goal. In addition, the Use Case Narratives are created from User Stories to describe how the system will accomplish what is described in the user story. There is a food supplier network. The resort also makes its own hot-sauce. These types of activities are supported through the automated features, like creating weekly work schedules based upon scheduled entries done by supervisors and the expected need.

The application will be developed in Visual Studio 2017 version 4.61. Framework libraries will be deployed for window 7 development using wpf. MySQL server will be used for data back end data management. We will have 4 three-week sprints, with in between weeks for recovery and discovery of backlog item to be finished in order to move forward on the next sprint. The ultimate goal is to provide Millennial Resorts with a great management and information system. The first step toward that goal was to analyze the current situation. Last Resort Software uses Scrum as our agile methodology to incrementally implement solutions. We used that methodology to analyze the current situation using modern design tools.

This report is a compilation of that work. Because of this extensive analysis of the business needs of Millennia Resorts we are confident that we will be able to deliver a final product that suits the needs of the resort's guests, employees and investors. The next step is to build and implement the solution based upon our analysis.

## Technology Requirements

Develop in Visual Studio 2017 version 4.61. Framework libraries will be deployed for Windows 7 development using wpf. MySql server will be used for data back end data management. The client applications will be built for Windows 7 above with scalable user interfaces and we will be using ASP.net for online content. Data tier application will be based on client server relationship for desktop. There will be a client to server relation for the front-end application using Chrome and Firefox for web testing and versioning for front-end development. We will use Sql 2012 or newer software. We will augment the wpf with exceeds ui toolkit community edition. Web app is will use the standard jQuery and ui libraries. Unit testing will use built in testing for windows applications. Hand coded mocks and fakes will be used for testing. We will manage project with a GitHub shared repo that we will be added to as contributors. Trello and Google Docs will also be used to manage code content. Deployment will be to a web server and database server. Any standard pc workstation. Supported by network IT. Security will be provided using Microsoft identity system form users will be in separate system, in order to allow the payment systems to pull data.

## Expected Timeline

Jan 14, 2019 -

### Spring Semester 2019

JANUARY							FEBRUARY							MARCH							APRIL							MAY																				
S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S														
							①	2	3	4	5				1	2					1	2						1	2	3	4																	
6	7	8	9	X	X	12		3	4	5	6	7	8	9	10	11	12	13	14	15	16	10	12	13	14	15	16	14	15	16	17	18	19	20	1	2	3	4										
13	14	15	16	17	18	19		17	18	19	20	21	22	23	24	25	26	27	28	29	30	24	25	26	27	28	29	30	21	22	23	24	25	26	27	5	6	7	8	9	10	11						
20	21	22	23	24	25	26		24	25	26	27	28			31							28	29	30					12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
27	28	29	30	31																																												

We will have 4 three week sprints, with in between weeks for recovery and discovery of backlog item to be finished in order to move forward on the next sprint. The sprints will be short, time-boxed periods when a Kirkwood scrum teams work to complete a set amount of work. Sprints are at the very heart of scrum and agile methodologies, and getting sprints right will help your agile team ship better software with fewer headaches.

## Actor List and User Stories

### Actor List

#### Generic

Admin	(Has access to everything. They can CRUD things nobody else can)
Manager	(Has Access to Every Role and Department Function)
Supervisor	(Has Access to Scheduling for their Department + all Employee things)
Worker	(Has Access to Generic Worker stuff along with their Roles and Department)
Guest	(Has Access to everything related to their account)
Dependent	(Is either a Child or a Pet. This is here for reference. They have no User Stories)
Public User	(Has access to what is available on our website)

#### Specific Role

Food Preparer	(Includes Bartenders, Cooks, Chefs)
Server	(Includes Hosts)
Valet	
Receptionist	(Includes Front Desk and Sales Clerk)
Doctor	(Includes Vet)
Talent	
Shuttle Driver	
Groomer	(Includes Massage Therapists, Barbers, and any skilled laborer by Appointment)

#### Specific Department

Inventory	(Generic)
Events	(Food Preparer, Talent, Generic)
Sanitation	(Generic)
Resort Ops	(Valet, Receptionist, Doctor, Shuttle Driver, Security, Generic)
Child Care	(Generic)
Pet Care	(Generic)
Maintenance	(Generic)

Food (Server, Food Preparer, Generic)  
 Marketing (Generic)

Primary Actors can be as specific or generic as needed. We can say every Worker can do something or only Marketing Managers can do something.

The catch is that EVERY person defined as a Primary Actor has to be able to do the Use Case.

For example, who can Delete an Event? A Food Manager cannot. An Event Supervisor and an Event Worker cannot Delete an Event either so there is one Primary Actor. Event Manager can Delete an Event.

## User Stories

The following User Stories are the gathered activities that each user wants to accomplish and what they need the software to do to accomplish that activity.

Use Case ID	Feature	Actor	Description	Outcome	Author
1.1.03	Create New Budget Report	Manager	Create New Budget Report	To Create A New Budget Report (Monthly/Quarterly/Yearly/Etc.)	Brandon Beltz
1.1.05	Create New Recipes	Food Preparer	Create New Recipes	I Can Add More Recipes	Brandon Beltz
1.1.06	Create New Time Off Request	Worker	Create New Time Off Request	Manager Can Approve Or Deny It	Brandon Beltz
1.1.08	Create Inventory Order	Inventory Manager	Add Orders To Stock Inventory	So I Can Add More Items To Stock	Chase Schulte
1.1.09	Create Dishes	Food Preparer	Create Dishes And Add Records	I Can Create New Dishes And Add It To System/Menu	Ramesh Adhikari
1.1.11	Provide Information About The Restaurant	Manager	Create Insightful Information To Customers About The Restaurant	So I Can Help The Customer Understand The Restaurant	Chase Schulte
1.1.16	Create Ticket For Table	Food Worker	Create Tickets	I Can Store Purchases Made Through Each Day	Richard Carroll
1.1.19	Create New Employee Roles	Manager	Create New Employee Roles	Assign/Add A New Role To An Active Employee	Brandon Beltz
1.1.20	Create New Work Schedule	Supervisor	Create New Work Schedule	Write New Weekly Schedule For Employees	Brandon Beltz
1.2.01	Log In	User (Internal)	Log In	I Can Access The Options Available To My Role	Richard Carroll
1.2.02	Log Out	User (Internal)	Log Out	I Can Get Off Work	Richard Carroll
1.2.08	Retrieve Budget Reports	Manager	Retrieve Budget Reports	Review Budget Reports	Brandon Beltz
1.2.14	Retrieve Recipes	Food Preparer	Retrieve (Browse) Recipes	Know What Ingredients To Use / How To Make	Brandon Beltz
1.2.18	Check Inventory Of Items	Inventory Manager	Update Inventory Of Items	So I Can See Current Stock In Inventory	Chase Schulte

Use Case ID	Feature	Actor	Description	Outcome	Author
1.2.21	View Sold Tickets	Food Preparer	Retrieve The Sold Tickets	I Can View The Tickets Sold And Send It To Manager/Quantity Of Item Sold	Ramesh Adhikari
1.2.28	Browse Orders	Food Preparer	View Orders I Need To Make	So I Know What To Make Next	Richard Carroll
1.2.29	View Recipe	Food Preparer	View Recipe Details	So I Know How To Make Dishes I Am Unfamiliar With	Richard Carroll
1.2.36	View Open Hours Of Restaurant/Hotel	Customer	View Hours Of Operation For The Restaurant	I Know When The Restaurant Is Open For Business	Phil Hansen
1.2.39	View Products	Customer	View Available Products	Find Products I'm Interested In	Richard Carroll
1.2.40	Review Cart	Customer	Review My Cart	Proceed To Purchase Items	Richard Carroll
1.2.45	Retrieve Employee Roles	Manager	Retrieve Employee Roles	Look Up/Browse Employee Roles	Brandon Beltz
1.2.49	Retrieve Employee Schedule	Manager	Retrieve Employee Schedule	Read Schedule	Brandon Beltz
1.2.50	Browse Employee Schedule	Manager	Browse Schedule	Browse Schedules	Brandon Beltz
1.3.06	Update Budget Reports	Manager	Update Budget Reports	Make Changes To Budget Reports	Brandon Beltz
1.3.07	Add Food Items and/or Beverages To Ticket	Server	Add Food/Beverage Items To The Ticket	I Can Add A Total For The Table's Bill, And Be Ready To Send It To The Kitchen	Phil Hansen
1.3.10	Update Recipes	Food Preparer	Update Recipes	Change Recipe Ingredients, Steps, Etc.	Brandon Beltz
1.3.13	Update Quantities To Correct Mistakes	Inventory Manager	Update Quantities Manually In Inventory	So I Can Manually Update Stock That Wasn't Automatically Done	Chase Schulte
1.3.20	Checkout	Customer	Checkout My Items	Obtain My Items	Richard Carroll
1.3.23	Update Employee Roles	Manager	Update Employee Roles	Change Employee's Assigned Role	Brandon Beltz
1.3.24	Update Employee Schedule	Manager	Update Employee Schedule	Make Changes To An Existing Schedule	Brandon Beltz
1.4.03	Delete Budget Reports	Manager	Delete Budget Reports	Delete Budget Reports	Brandon Beltz
1.4.05	Delete Recipes	Food Preparer	Delete Recipes	Remove Recipes We Will No Longer Make	Brandon Beltz
1.4.06	Remove Items From Stock	Inventory Manager	Deactivate Items In Inventory	So I Can Remove A Quantities Of Items From Stock	Chase Schulte
1.4.07	Cancel Orders	Inventory Manager	Deactivate Orders In Progress	So I Can Cancel Incoming Orders That Aren't Needed	Chase Schulte
1.4.08	Remove Dishes From Menu	Food Preparer	Deactivate Dishes From Menu	Deactivate Dishes From Menu If It Recalled/Customers Safety	Ramesh Adhikari
1.4.13	Delete Employee Roles	Manager	Delete Employee Roles	Delete Employee's Assigned Role	Brandon Beltz
1.4.15	Delete Employee Schedules	Manager	Delete Employee Schedules	Archive Old Schedules	Brandon Beltz
2.1.02	Add Drink	Food Preparer	Add A New Drink	A New Drink Is In The Db	Jesse Tomash
2.1.09	Create New Menu Item	Chef	Create A New Menu Item	A New Item Has Been Created For The Menu	Craig Barkley

Use Case ID	Feature	Actor	Description	Outcome	Author
2.1.14	Add Items To Inventory	Manager	Add New Items To Inventory	Purchase And Track New Items In Inventory	Kevin Broskow
2.1.15	Create Supply	Inventory Personnel	Make A Supply Order	Our Inventory Can Be Replenished	Dalton Cleveland
2.1.16	Create Maintenance Ticket	Manager	Send Tickets For Repairs To Maintenance	Have Them Fix Problematic Issues	Kevin Broskow
2.1.17	View Recipe Details	Admin	Read A Menu In The System.	I Can Read Menu Choices In Our List	Eduardo Colon
2.1.18	Create New Building	Manager	Create New Building	New Building Has Been Added To The Database	Craig Barkley
2.1.21	Create Report	Admin	Create A Product Report In The System	I Can Handle To The Sales Department	Eduardo Colon
2.1.24	Create Order From Menu	Inventory Personnel	Create An Order From The Menu	So Our Product Can Be Replenished	Dalton Cleveland
2.1.25	Create Resort Service	Manager	Create A New Resort Service	The New Resort Service Has Been Added	Craig Barkley
2.1.27	Create New Set List	Manager	Create A New List Of Sets	The List Of Sets Has Been Created	Craig Barkley
2.2.01	Browse/Search Menu List	Admin	Check A Product In The System	I Can Replenish It	Eduardo Colon
2.2.05	View Recipe Details	Food Preparer	View Details For A Specific Recipe	View Ingredients, Processes And Times	Kevin Broskow
2.2.06	View Current Orders	Food Preparer	View Orders For Current Event	Prepare All Food Items	Kevin Broskow
2.2.10	Search Menu List	Admin	Search The Menu List	Display The List Of Menus	Craig Barkley
2.2.17	View Recipe Details	Manager	View Recipe Detail	I Can See What Is In The Recipe	Craig Barkley
2.2.18	Search Event Schedule	Maintenance Person	Search An Event Schedule In The System	I Can Prepare The Room For That Event	Eduardo Colon
2.2.20	Read Event Request	Maintenance Person	To Check An Event Schedule In The System	I Can Plan For Any Other Activity	Eduardo Colon
2.2.22	Search Item	Web Visitor	Search An Item For A Menu In The System	I Can Add It To My Chosen List	Eduardo Colon
2.2.24	Search Order Supplies	Inventory Personnel	Search Order Supplies In The System	I Can Narrow My Search Criteria.	Eduardo Colon
2.2.26	Browse Order Supplies	Inventory Personnel	Browse Orders In The System	I Can Choose An Order And Modify It.	Eduardo Colon
2.2.28	Read Supply	Manager	Read Supply List	I Have Located The Supply Item I Wish To Edit	Eduardo Colon
2.2.34	View Maintenance Work Orders	Manager	View Current Maintenance Work Orders	Track Their Status	Kevin Broskow
2.2.50	Read Catering Request Details	Manager	See The Requests	I Can See Requests And Manage Them.	Craig Barkley
2.2.54	Read Supplier	Marketing	I Need To View Authorized Suppliers	So That I Can Contact Authorized Suppliers For Needs In A Marketing Event.	Carlos
2.2.56	Read List Of Buildings	Manager	Get A List Of Buildings	So That I Can Manage Automation	Craig Barkley
2.2.58	Read Package Detail	Manager	Get A List Of Buildings	The Packages Details Are Displayed	Craig Barkley

Use Case ID	Feature	Actor	Description	Outcome	Author
2.2.60	Browse Supply Details	Vendor	Browse The Supply Details	The Supply Details Are Displayed	Craig Barkley
2.3.11	Change Drink/Recipe	Food Preparer	Edit Drink Recipes	Drink Recipes Are Current	Jesse Tomash
2.3.16	Edit Menu List	Food Preparer	To Adjust Menu Items	I Can Update Available Offerings	Craig Barkley
2.3.20	Update Resort Service	Manager	Update The Information About Our Resort Services	So That Our Customers Can Accurately See The Services That Our Resort Provides	Dalton Cleveland
2.3.21	Edit Product Report	Admin	Update The Information In A Product Report	So That The Information Can Be Current And Accurate	Dalton Cleveland
2.3.22	Edit Order Supplies	Inventory Personnel	Edit Order Supplies In The System	I Can Add Or Remove Chosen Item From It	Eduardo Colon
2.3.24	Update Customer Order	Customer	Update An Active Order That I Have Placed	So That I Can Remove Any Unwanted Items, Or Add Something That Was Missed	Dalton Cleveland
2.3.27	Update Package Details.	Manager	Update A Package	The Packages Details Are Displayed	Craig Barkley
2.3.33	Update Building Details.	Manager	Update Building Details.	Updates Details Are Displayed	Craig Barkley
2.4.07	Delete Event Review	Admin	Delete Event Review	The Event Review Has Been Deleted	Craig Barkley
2.4.08	Deactivate Menu List	Admin	Deactivate A Menu In The System.	I Can Remove It, If It Is No Longer Available	Eduardo Colon
2.4.10	Deactivate Customer Vehicle	Admin	Deactivate A Customer Vehicle	The Customer Vehicle Has Now Been Deactivated	Eduardo Colon
2.4.11	Delete Comments	Manager	Delete An Inappropriate Or Unnecessary Comment In The System	We Can Maintain A Friendly And Ethical Work Environment	Dalton Cleveland
2.4.12	Delete Building	Admin	Delete A Building	The Building Has Now Been Deleted	Eduardo Colon
2.4.13	Delete Item	Web Visitor	Delete An Item From Menu In The System	I Can Decrease My Chosen List Items	Eduardo Colon
2.4.14	Cancel Order Supplies	Inventory Personnel	Cancel An Order Supply In The System	I Can Avoid Any Charges In No Longer Needed Orders	Eduardo Colon
2.4.50	Delete Resort Service	Manager	Delete A Resort Service From Resort	I Have An Item Highlighted And Selected A Button Labeled Delete Item	Craig Barkley
3.1.03	Create Event	Event Manager	I Need To Create An Event	So That The Event Can Be Added To The Schedule	Matt
3.1.04	Create Event Request	Event Manager	I Would Like To Create An Event Request	So That I Can Host An Event	Caitlin
3.1.05	Create Catering Request	Event Manager	I Need To Create A Catering Request	So That I Can Create A Catering Event.	Caitlin
3.1.06	Create Advertising Event	Marketing	I Need To Create An Event Or Service For Promotional Stands	So That I Can Create And Event	Carlos
3.1.08	Create New Sponsor	Marketing	I Need To Book Sponsors	So That I Can Coordinate Sponsors With The Event Schedule	Caitlin

Use Case ID	Feature	Actor	Description	Outcome	Author
3.1.09	Create Packages	Marketing	I Need To Create Packages	So That We Can Create Special Packages For Specific Customers Or Companies.	Carlos
3.1.10	Create Account	Talent	I Would Like To Create An Account	So That I Can Perform At An Event	Jacob
3.1.16	Create New Supplier	Marketing	I Need To Add Outsourcing Suppliers	So That I Can Contact Authorized Outsourcing Suppliers For Needs In A Marketing Event.	Carlos
3.1.17	Create Performance	Talent	I Would Like To Add My Performances To My Account	So That I Can Perform At The Event	Jacob
3.1.50	Create Review	Customer	I Would Like To Leave A Review For An Event That I Have Attended	So That Other People Can See What Events Are Good/Bad	Caitlin
3.2.19	Read Food Menu	Marketing	I Need To View The Food Service Menu For An Event	So That We Can Advertise The Normal, Seasonal And Local Menus From Our Restaurant And Catering Services.	Carlos
3.2.21	Read Budget	Marketing	I Need To View The Department Budget	So That I Can View The Budget For Marketing For Planning Marketing Events.	Carlos
3.2.22	Read Supplier	Marketing	I Need To View Authorized Suppliers	So That I Can Contact Authorized Suppliers For Needs In A Marketing Event.	Carlos
3.2.23	Read Advertising Event	Marketing	I Need To View Promotional Items	So That We Can See What Promotional Items We Have Available.	Carlos
3.2.25	Browse Suppliers	Inventory	I Need To View Suppliers	So That We Can Have The List Of Authorized Suppliers.	Carlos
3.2.30	Read Resort Services	Web Visitor	I Would Like To See The Services The Resort Provides	So That I Can Decide To Attend The Resort	Carlos
3.2.31	Read Event Request	Event Host	I Would Like To See My Event Request Details	So That I Can See What I Need To Change	Matt
3.2.32	Read Building List	Event Host	I Would Like To See A List Of Buildings	So That I Can Reserve The Building I Want To Host An Event At	Matt
3.2.34	Read Review	Customer	I Would Like To See Reviews For Past Events	So That I Can See What Events Are Good/Bad	Matt
3.2.35	Read Event	Customer	I Would Like To See The Details For An Event	So That I Know What Is Happening At The Event	Jacob
3.2.36	Read Setup List	Event Staff	I Need To See The List Of What To Set Up	So That I Can Set Up The Correct Amount For An Event	Matt
3.2.37	Read Performance Details	Talent	I Would Like To Read Through My Performance Details	So That I Know What Is In My Performance	Caitlin
3.2.38	Read account details	Talent	I Would Like To Read Through My Account Details	So That I Know Exactly What Is In My Account	Caitlin

Use Case ID	Feature	Actor	Description	Outcome	Author
3.2.44	Browse Event Request	Event Manager	I Would Like To See A List Of All Our Event Requests	So That I Can See Which Event Requests	Matt
3.2.45	Browse Event	Customer, Worker	I Would Like To See A List Of All Our Events	So That I Can See Which Events	Matt
3.2.46	Browse Performance	Customer, Worker	I Would Like To See A List Of All Our Performances	So That I Can See Which Performances	
3.2.47	Browse Resort Service	Customer, Worker	I Would Like To See A List Of All Our Resort Services	So That I Can See Which Resort Services	
3.2.48	Browse Account	Manager	I Would Like To See A List Of All Our Accounts	So That I Can See Which Accounts	Austin B
3.2.49	Browse Advertising Event	Marketing, Manager	I Would Like To See A List Of All Our Advertising Events	So That I Can See Which Advertising Events	Austin B
3.2.50	Browse Package	Marketing, Customer	I Would Like To See A List Of All Our Packages	So That I Can See Which Packages	Matthew Hill
3.2.51	Browse RSVP	Event Host, Event Manager	I Would Like To See A List Of All Our RSVPs	So That I Can See Which RSVPs	Caitlin
3.2.52	Browse Setup List	Worker	I Would Like To See A List Of All Our Setuplists	So That I Can See Which Setuplists	Caitlin
3.2.53	Browse Review	Customer, Worker, Manager	I Would Like To See A List Of All Our Reviews	So That I Can See Which Reviews	Dalton
3.3.06	Update RSVP	Customer	I Would Like To Update My Rsvp	So That If Something Changes In My Plans I Could Update Or Change My Reservation.	Caitlin
3.3.13	Update Catering Request	Event Manager	I Need To Update A Catering Request	So That I Can Add Or Remove Catering Needs.	Caitlin
3.3.15	Update Advertising Event	Marketing	I Need To Edit Event Or Service For Promotional Stands	So That I Can Edit The Event's Details. Add Or Remove.	Carlos
3.3.19	Update Supplier	Marketing	I Need To Edit Outsourcing Suppliers	So That We Can Edit The Information Of A Outsourcing Suppliers.	Carlos
3.3.20	Update Package	Marketing	I Need To Edit Packages	So That We Can Add Or Remove Features From The Packages.	Carlos
3.3.26	Update Performance	Talent	I Would To Edit My Performances To My Account	So That I Can Request Different	Jacob
3.3.27	Update Scheduled Performance	Talent	I Would Like To Update My Scheduled Performances	So That My Performance Details Are Accurate For An Event	Jacob
3.3.29	Update Account	Web Visitor	I Would Like To Edit My Account	So The Details My Account Can Be More Accurate	Jacob
3.3.30	Update Event Request	Event Host	I Would Like To Update An Event Request I Created	So That The Event I Created Is How I Wanted It	Caitlin

Use Case ID	Feature	Actor	Description	Outcome	Author
3.3.31	Update Event Details	Event Host	I Would Like To Update An Approved Events Details	So That The Event Can Work How I Want It To	Jacob
3.4.02	Delete Catering Request	Event Manager	I Need To Delete A Catering Request	So That I Can Cancel A Catering Event.	Caitlin
3.4.03	Delete Advertising Event	Marketing	I Need To Cancel Event Or Service For Promotional Stands	So That I Can Cancel The Event, Noting The Reason.	Carlos
3.4.05	Delete Supplier	Marketing	I Need To Delete Outsourcing Suppliers	So That We Can Delete A Outsourcing Supplier That Is Not Going To Work Anymore With The Resort.	Carlos
3.4.06	Delete Package	Marketing	I Need To Delete Packages	So That We Can Delete A Package.	Carlos
3.4.09	Delete Performance	Talent	I Need To Delete My Performances From My Account	So That I Am Not Expected To Attend An Event	Jacob
3.4.10	Delete Scheduled Performance	Talent	I Need To Cancel Scheduled Events	So That The Events Are No Longer On The Schedule	Jacob
3.4.11	Delete Account	Web Visitor	I Would Like To Request Deactivation Of My Account	So That I Cannot Book A Resort On Accident	Jacob
3.4.12	Delete Event Request	Event Host	I Would Like To Delete My Event Request	So That The Event Request Does Not Get Approved	Matt
3.4.13	Delete RSVP	Customer	I Would Like To Cancel My Rsvp	So That I Can Let The Host Know In Advance That I Will No Longer Be Able To Attend The Event.	Caitlin
3.4.14	Delete Events	Event Manager	I Need To Cancel An Event	So That The Event Is No Longer On The Schedule	Caitlin
4.1.04	Create Product	Inventory Control	Create New Records For Products	Inventory Control Can Keep Track Of New Products That Are Being Made.	Jared
4.2.04	View Product	Inventory Control	View Information About A Product	I Can Review Product Information.	Jared
4.2.17	Search For Supplier Account	Employee	Search For A Specific Supplier Account	Easily Access Contact Information And Products So I Can Place Orders	Dani
4.2.37	Browse Supply Requests	Manager	Browse A List Of Open Supply Requests	Increase Efficiency By Consolidating Similar Orders	Dani
4.2.41	Browse Items	Customer	Browse Items	I Can See What I Could Like To Purchase	Dani
4.3.04	Update Supply Order	Manager	Update The Supply Order	Supply Order List Has Been Updated	Jared Greenfield
4.3.05	Edit Product	Inventory Control	Edit The Information About A Product	Product Information Can Stay Current And Effective.	Jared
4.3.21	Update Work Schedule	Manager	Update A Work Schedule	Our Employees Can Work Their Shift According To Their Availability	Dalton Cleveland
4.4.02	Deactivate Product	Inventory Control	Deactivate Product	Inventory Control Can Stop Keeping Records Of Products That Are No Longer In Production	Jared
4.4.03	Delete Supply Order	Manager	Delete A Supply Order	Supply Order Is Deleted	Dani Russo
4.4.05	Delete Work Schedule	Production Scheduler	Delete A Work Schedule	Work Schedule Has Been Deleted	Austin Berquam

Use Case ID	Feature	Actor	Description	Outcome	Author
5.1.02	Add A Reservation	Receptionists	Add A Reservation	A New Reservation Is In The List	Wes Richardson
5.1.06	Add Guest's Vehicle	Valet	Add A Vehicle Record	I Can Store Details About A Guest's Car	James Heim
5.1.07	Create Hotel Room	Manager	Create A Hotel Room	So That We Can Rent Out The Room To A Guest	Craig Barkley
5.1.08	Create Shuttle Vehicle	Shuttle Driver	Add A Vehicle Record	The Vehicle May Be Used As A Shuttle	James Heim
5.1.13	Schedule Booking	Guest	Schedule My Booking Information.	I Can Extend My Stay At The Resort.	Matthew Hill
5.2.01	View Details Of Reservation	Receptionists	View Details Of Reservation	See The Details Of A Reservation	Wes Richardson
5.2.05	View List Of Maintenance Tickets	Receptionists	View List Of Maintenance Tickets	View A List Of The Maintenance Tickets	Wes Richardson
5.2.06	View A List Of Reservations	Receptionists	View A List Of Reservations	See All The Reservations	Wes Richardson
5.2.07	View A List Of Rooms	Receptionists	View A List Of Rooms	See A List Of The Rooms	Wes Richardson
5.2.08	View Maintenance Ticket Details	Receptionists	View Maintenance Ticket Details	See The Details Of A Maintenance Ticket	Wes Richardson
5.2.09	View Room Details	Receptionists	View A Room Details	See The Details Of A Room	Wes Richardson
5.2.10	View Supply Order Detail	Supply Order	View Supply Order Detail	View The Details Of A Supply Order	Wes Richardson
5.2.27	Browse Guest Vehicles	Valet	Browse Guest Vehicles	I Can Retrieve Details About A Guest's Car	James Heim
5.2.28	Read Guest Vehicles	Valet	Read Guest Vehicle	I Can View All Cars Registered With A Guest	James Heim
5.2.37	Read Shuttle Vehicle By Vin	Shuttle Driver	Retrieve Shuttle Vehicle By Vin	I Can View The Details Of A Shuttle Vehicle	James Heim
5.2.38	Retrieve Shuttle Vehicle By License Plate	Shuttle Driver	Retrieve Shuttle Vehicle By License Plate	I Can View The Details Of A Shuttle Vehicle	James Heim
5.3.01	Check Guest In	Receptionists	Check Guest In	Check In A Guest That Did Not Have A Reservation	Wes Richardson
5.3.06	Edit A Reservation	Receptionists	Edit A Reservation	The Reservation Has Been Changed	Wes Richardson
5.3.12	Update Vehicle	Valet	Update A Vehicle's Record	We Can Keep Track Of A Guest's Vehicle	James Heim
5.3.15	Update Maintenance Ticket	Shuttle Driver	Update The Progress Of A Maintenance Ticket	I Can Close A Maintenance Ticket	James Heim
5.3.16	Update Shuttle Vehicle Status	Shuttle Driver	Update A Shuttle Vehicle Status	I Can Mark A Vehicle As Being In Available, In Use, In Service, Or Decommissioned	James Heim
5.3.18	Update Hotel Room	Receptionists	Update A Hotel Rooms Reservation	The Guest Can Be In Their Desired Room, And Update Each Rooms Availability	Dalton Cleveland
5.4.02	Deactivate A Reservation	Receptionists	Deactivate A Reservation	Delete The Reservation	Wes Richardson
5.4.03	Deactivate Maintenance Ticket	Maintenance Worker	Deactivate Maintenance Ticket	I Can Inform Other Workers That A Slip Is Completed, And No Longer Needs Maintenance	Dalton Cleveland

Use Case ID	Feature	Actor	Description	Outcome	Author
5.4.05	Deactivate Shuttle Vehicle	Shuttle Driver	Deactivate A Shuttle Vehicle Record	I Can Remove A Vehicle From Our Inventory	James Heim
5.4.06	Deactivate Room	Janitor	Deactivate A Room	Report A Room's Status As Not In Service / Not Currently Habitable.	Matthew Hill
6.1.03	Register	Guest	Register For An Event	I Can Have My Child Go To The Event	Tiona White
6.1.06	Create User Account	Guest	Create User Account	I Can Use Child Care System For Guests And Students.	Gunardi Saputra
6.1.08	Create A Message (From Staff)	Staff	Create A New Message To The Manager, Staff And Guests	I Can Have Feedback From The Manager, Staff And Guests	Gunardi Saputra
6.2.05	Browse Children	Manager	See List Of Children In Daycare Or In Events	I Know Everything That Is Going On And What My Staff Needs To Do	Tiona White
6.2.06	View Child Details	Manager	See The Childs Information	I Can See Age, Guest Contact Information, Etc.	Tiona White
6.3.23	Update Dependent Record	Guest	Update Dependent Information	So That He/She Could Have Updated Information	Francis Mingomba
6.4.03	Cancel Registration	Guest	Cancel My Registration	The Resort Will Know Not To Accommodate An Extra Person For Their Event	Dalton Cleveland
6.4.04	Deactivate Guest Account	Manager	Deactivate Guest Account	I Can See Active And Inactive Accounts	Gunardi Saputra
7.1.03	Create Groomer Appointment	Pet Staff	Sign Up For Appointments	A Record Has Been Created For The Pet To Have An Appointment With Groomer	Austin Delaney
7.1.04	Add Medical Record	Pet Worker	Add A Pet's Medical Information Record	The Pet's Medical Record Is Now Recorded	Austin Delaney
7.2.01	Read Dependent Record	Pet/Child Worker	View The Details Of A Dependent's Record	I Can Fulfill Related Services For The Guest	Ben Hanna
7.2.14	Browse Groomer Appointments	Groomer	Look At Appointment Book	Groomer Book Displayed	Cody Herb
7.2.16	Read Details Of Appointment	Groomer	Details For Appointment	Details Of Groomer Book Displayed	Cody Herb
7.2.21	Browse Medical Info	pet Staff	Browse Medical Info	I Can Find A Number Of Pets By Activity	Ben Hanna
7.2.33	Read Medical Info	Vet tech	Pull A Pet's Medical Information	A Pet's Medical Information Is Displayed To User	Austin Delaney
7.3.01	Update Dependent Record	Customer	Update My Personal Record	So The Resort Has The Correct Information If They Need To Contact Me.	Dalton Cleveland
7.3.04	Update Pet Medical Info	Vet tech	Update A Pet's Medical Info	Pet Medical Information Is Updated	Alisa Roehr
7.3.14	Update Appointment Details	Pet Receptionist	Change An Appointment Details	Account For A Change In Schedule.	Ben Hanna
7.3.15	Update Medical Requests	Pet Receptionist	Add Medical Requirements Or Special Treatment Requests	Pet Medical Requirements Or Special Treatments Are Updated	Alisa Roehr
7.4.02	Delete Dependent Record	Manager	Delete Dependent Record	Delete Dependent Record In The System	Cody Herb
7.4.05	Cancel/Delete Appointment	Pet Receptionist	Delete/Cancel Appointments	Appointment Canceled/Deleted	Cody Herb

## Use Case Narratives

The following Use Case Narratives are created from User Stories to describe how the system will accomplish what is described in the user story. It will include what a typical workflow, alternate flows and exceptions that the system may encounter while doing the task, what changes are made to the system, and what place in the system the user will be at when they finish the task. They are numbered based on area of the Resort, Database functionality and order they were created.

### Food Operations

The following use cases detail the design relating food operations and items.

#### 1.1.05 Create Recipes

<b>Use Case ID:</b>	1.1.05		
<b>Use Case Name:</b>	Create Recipes		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Brandon Beltz
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-5
<b>Actors:</b>	Primary: Food Preparer		
<b>Description:</b>	Create new recipes in the database		
<b>Trigger:</b>	User requests system to create a new recipe		
<b>Preconditions:</b>	1. The Assumptions are true 2. Recipe database must exist 3. If so, recipe database must be accessible (not locked)		
<b>Postconditions:</b>	1. A new recipe is created in the database		
<b>Normal Flow:</b>	1. User chooses to create new recipe 2. System prompts user to complete new recipe form 3. Once all required fields are entered, user selects submit 4. New record is created in the recipes database		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the User is not logged in to the system 1. System will prompt User to log in 2. User enters correct login information 3. Use Case resumes on step 2 2a. In step 2 of the normal flow, if the recipe database is not found or not accessible 1. System will alert user that database was not found or is locked by someone/is unavailable 2. Use Case terminates		

	<p>3a. In step 3 of the normal flow, if the recipe already exists in the database</p> <ol style="list-style-type: none"> <li>1. System will alert user that the record already exists in the database</li> <li>2. Use Case terminates</li> </ol>
<b>Exceptions:</b>	<p>1b. In step 1 of the normal flow, if the User enters incorrect login information</p> <ol style="list-style-type: none"> <li>1. System will alert that login credentials were incorrect and prompt User to re-enter login information</li> <li>2. Step repeats until valid credentials are entered</li> <li>3. Once entered, Use Case resumes on step 2 of normal flow</li> </ol>
<b>Includes:</b>	System
<b>Frequency of Use:</b>	On demand, typically 1 to 50 times per day
<b>Special Requirements:</b>	NA
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	NA

### 1.1.09 Add Recipes

<b>Use Case ID:</b>	1.1.09		
<b>Use Case Name:</b>	Add recipes		
<b>Created By:</b>	Ramesh Adhikari	<b>Last Updated By:</b>	Austin Berquam
<b>Date Created:</b>	2018-09-11	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Chef Manager		
<b>Description:</b>	Chef manager wishes to add recipes		
<b>Trigger:</b>	Chef manager add recipes and saves the changes		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. An add recipes button exists in the system</li> <li>2. The Assumptions are true</li> </ol>		
<b>Postconditions:</b>	1. An add recipes will be displayed when user is logged into system		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The chef manager enters the system with login account</li> <li>2. The chef manager activates the “Add recipes” button in the system</li> <li>3. The system shows the fields or list to add the recipes</li> <li>4. The chef manager adds the recipes</li> <li>5. The system confirms if chef manager wants to add the recipes</li> <li>6. The chef manager saves the changes in system</li> <li>7. The chef manager logs out</li> </ol>		
<b>Alternative Flows:</b>	The chef manager selects “Add recipes” option		

	If the chef manager clicks cancel operation, the system stops  Chef manager enters wrong login email or password and need to go back
<b>Exceptions:</b>	1b. In step 1 of the normal flow, if the User enters incorrect login information <ul style="list-style-type: none"> <li>1. System will alert that login credentials were incorrect and prompt User to re-enter login information</li> <li>2. Step repeats until valid credentials are entered</li> <li>3. Once entered, Use Case resumes on step 2 of normal flow</li> </ul>
<b>Includes:</b>	System
<b>Frequency of Use:</b>	On demand, typically 1 to 50 times per day
<b>Special Requirements:</b>	To add recipes the system must be working
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 1.1.11 Provide information about the restaurant

<b>Use Case ID:</b>	1.1.11		
<b>Use Case Name:</b>	Provide information about the restaurant		
<b>Created By:</b>	Chase Schulte	<b>Last Updated By:</b>	Alisa Roehr
<b>Date Created:</b>	2018-09-09	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Customer, Manager		
<b>Description:</b>	Manager provides helpful information about the restaurant		
<b>Trigger:</b>	The Customer calls the Manager number		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Customer must have a means of contacting Manager via phone number</li> <li>2. Customer must want information about the restaurant</li> <li>3. The Assumptions are true.</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Customer is enriched with info about the restaurant</li> </ul>		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Customer calls</li> <li>2. Manager Picks up</li> <li>3. Customer asks about the restaurant</li> <li>4. Manager explains the restaurant</li> <li>5. Customer hangs up</li> </ul>		
<b>Alternative Flows:</b>	4a. In step 3 of the normal flow, if Manager is unable to explain something		

	<p>about the restaurant.</p> <ol style="list-style-type: none"> <li>1. Manager will redirect the call to another employee who is more knowledgeable in the problem</li> <li>2. Use Case Continues from step 2</li> </ol>
<b>Exceptions:</b>	<p>2-4a In step 2-4, if the Customer drops the call</p> <ol style="list-style-type: none"> <li>1. Customer calls back</li> <li>2. Customer is redirected to employee they were recently talking too</li> <li>3. Use Case Continues from step 2</li> </ol> <p>2a. If Manager is unable to pickup</p> <ol style="list-style-type: none"> <li>1. Customer will be left suspended in step 1 until a line is free</li> <li>2. Normal flow will continue</li> </ol>
<b>Includes:</b>	steps 1-2 and 4 are required in all flows pertaining to Manager
<b>Frequency of Use:</b>	80-100 times per Employee per day
<b>Special Requirements:</b>	Phone lines must be running
<b>Assumptions:</b>	<p>Both Customer and Manager speak the same language</p> <p>Both Customer and Manager</p> <p>User is logged in. User has permissions.</p>
<b>Notes and Issues:</b>	<p>1. How many calls will Manager be able to take</p>

#### 1.1.16 Create Food Ticket

<b>Use Case ID:</b>	1.1.16		
<b>Use Case Name:</b>	Create Food Ticket		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Austin Berquam
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Food Worker (Waiter, Bartender)		
<b>Description:</b>	Order system must make a ticket for the ordered items, so they can be purchased		
<b>Trigger:</b>	Waiter sends items requested by customer to the order system		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Ticket is not made yet</li> <li>2. The Assumptions are true</li> </ol>		
<b>Postconditions:</b>	1. Ticket is made		
<b>Normal Flow:</b>	1. Waiter sends items requested by customer to the order system		

	<ol style="list-style-type: none"> <li>2. Order System puts items together in a ticket</li> <li>3. Order system keeps ticket in storage to be retrieved by waiter when they need it</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. The Food Worker selects “Create Order” option</li> <li>2. If the user clicks cancel operation, the system stops</li> </ol>
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1b. In step 1 of the normal flow, if the User enters incorrect login information             <ol style="list-style-type: none"> <li>1. System will alert that login credentials were incorrect and prompt User to re-enter login information</li> <li>2. Step repeats until valid credentials are entered</li> <li>3. Once entered, Use Case resumes on step 2 of normal flow</li> </ol> </li> </ol>
<b>Includes:</b>	Create Ticket (Waiter)
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	To create orders the system must be working
<b>Assumptions:</b>	The Program displays text in the area’s native language. User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 1.2.14 Browse recipes

<b>Use Case ID:</b>	1.2.14		
<b>Use Case Name:</b>	Browse recipes		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Food Preparer		
<b>Description:</b>	Retrieve active recipes currently in the database		
<b>Trigger:</b>	User requests system to browse index of recipes		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User must be logged in</li> <li>2. Recipe database must exist</li> <li>3. If so, recipe database must be accessible (not locked)</li> </ol>		
<b>Postconditions:</b>	1. User gains access to the recipe database and can search for or browse recipes		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User chooses to retrieve recipes</li> <li>2. Recipe database is accessed</li> <li>3. User browses recipes / submits queries to the database</li> <li>4. Results of queries are displayed</li> </ol>		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the User is not logged in to the system		

<b>[Alternative Flow 1 – Not Logged In]</b>	<p>1. System will prompt User to log in      2. User enters correct login information      3. Use Case resumes on step 2</p> <p>1b. In step 1 of the normal flow, if the User enters incorrect login information</p> <ol style="list-style-type: none"> <li>1. System will alert that login credentials were incorrect and prompt User to re-enter login information</li> <li>2. Step repeats until valid credentials are entered</li> <li>3. Use Case resumes on step 2 of normal flow</li> </ol> <p>2a. In step 2 of the normal flow, if the recipe database is not found or not accessible</p> <ol style="list-style-type: none"> <li>1. System will alert user that database was not found or is locked by someone is unavailable</li> <li>2. Use Case terminates</li> </ol>
<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	Recipe database
<b>Frequency of Use:</b>	On demand, typically 1 to 50 times per day
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions.
<b>Notes and Issues:</b>	N/A

### 1.2.21 View Sold Tickets

<b>Use Case ID:</b>	1.2.21		
<b>Use Case Name:</b>	View Sold Tickets		
<b>Created By:</b>	Ramesh Adhikari	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-09-11	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Manager		
<b>Description:</b>	Restaurant Manager wishes to view sold tickets		
<b>Trigger:</b>	Restaurant Manager views the sold tickets from records		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. A view sold tickets menu exists in the system</li> <li>2. The Assumptions are true.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. A view sold tickets menu will be displayed</li> </ol>		

<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The Manager selects View Sold tickets</li> <li>2. The System responds by showing the list of sold tickets</li> <li>3. The Manager views the sold tickets from the record</li> <li>4. The Manager selects a ticket to display details</li> <li>5. The System displays details of specified ticket</li> <li>6. The Manager logs out</li> </ol>
<b>Alternative Flows:</b>	<p>2a) If there are currently no sold tickets in the system</p> <ol style="list-style-type: none"> <li>1. The System alerts the Manager that no sold tickets are currently in the system</li> <li>2. Use Case resumes on step 1 of Normal Flow</li> </ol>
<b>Exceptions:</b>	<p>2a) If the database query fails</p> <ol style="list-style-type: none"> <li>1. The System alerts the Manager and prompts them to try again</li> <li>2. Use Case resumes on step 1 of Normal Flow</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	<p>The User is a Manager of the Restaurant</p> <p>The User is logged in and has proper privileges.</p>
<b>Notes and Issues:</b>	N/A

#### 1.2.24 Search Recipes

<b>Use Case ID:</b>	1.2.24		
<b>Use Case Name:</b>	Search recipes		
<b>Created By:</b>	Ramesh Adhikari	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Chef Manager		
<b>Description:</b>	Chef manager wishes to search recipes		
<b>Trigger:</b>	Chef manager selects the option to search recipes		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. A search recipes button exists in the system</li> <li>2. The assumptions are true</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The recipe being searched for is retrieved</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user selects the option to search for a recipe</li> <li>2. The user enters the name for the recipe they are searching for</li> </ol>		

	<ul style="list-style-type: none"> <li>3. The system looks through the list of recipes and returns a list of recipes relating to the keyword that the user entered</li> <li>4. The user selects the recipe they were looking for</li> <li>5. The system returns the details of the selected recipe</li> <li>6. The user reads the recipe</li> <li>7. The user closes the window and returns the user back to the previous page.</li> </ul>
<b>Alternative Flows:</b>	None currently.
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>3a. The system cannot find the recipe the user is searching for           <ul style="list-style-type: none"> <li>1. The system alerts the user that no matching recipes were found</li> <li>2. The user acknowledges the systems prompt.</li> <li>3. The system returns the user back to the previous page</li> </ul> </li> <li>5a. The system cannot find the details for the selected recipe           <ul style="list-style-type: none"> <li>1. The system alerts the user that no details were found for the selected recipe</li> <li>2. The user acknowledges the systems prompt.</li> <li>3. The system returns the user back to the previous page</li> </ul> </li> </ul>
<b>Includes:</b>	Create, Update, Delete Recipe
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	None currently
<b>Assumptions:</b>	The user is logged in. The user has permissions.
<b>Notes and Issues:</b>	None currently

#### 1.2.28 Browse Orders

<b>Use Case ID:</b>	1.2.28		
<b>Use Case Name:</b>	Browse Orders		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Brandon Beltz
<b>Date Created:</b>	2018-09-10	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Food Preparer		
<b>Description:</b>	Browse a list of orders placed		
<b>Trigger:</b>	User requests to browse orders placed		
<b>Preconditions:</b>	1. The Assumptions are true 2. User has access to food service options		

<b>Postconditions:</b>	1. Order list is displayed for browsing
<b>Normal Flow:</b>	1. User requests to browse order list from 2. Program retrieves list from Internal Order System 3. Program displays order list to the User
<b>Alternative Flows:</b>	1. Food list is empty (Normal flow step 2) a. Program displays message saying the food list is empty b. Program returns to Normal flow step 1
<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	Order System
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The Program displays text in the area's native language. User is logged in. User has permissions.
<b>Notes and Issues:</b>	1. Should this be a data action?

#### 1.2.29 View Recipes Details

<b>Use Case ID:</b>	1.2.29		
<b>Use Case Name:</b>	View Recipe Details		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-09-11	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Food Preparer		
<b>Description:</b>	View details of a selected recipe		
<b>Trigger:</b>	User requests to view details of a selected recipe		
<b>Preconditions:</b>	1. User is logged in 2. Recipe database must exist 3. If so, recipe database must be accessible (not locked) 4. List of recipes has been browsed (1.2.14)		
<b>Postconditions:</b>	1. Recipe details are displayed		
<b>Normal Flow:</b>	1. User requests to view details of a selected recipe 2. Program retrieves recipe details 3. Program displays recipe details to the User		
<b>Alternative Flows:</b>	2a) If selected recipe has no details currently in the database 1. System alerts User that selected recipe currently has no details associated with it 2. Use Case resumes on step 1 of Normal Flow		

<b>Exceptions:</b>	2a) Database Query fails <ol style="list-style-type: none"> <li>1. System alerts User that Database Query has failed</li> <li>2. Use Case resumes on step 1 of Normal Flow</li> </ol>
<b>Includes:</b>	1.2.14 Browse Recipes
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The Program displays text in the area's native language
<b>Notes and Issues:</b>	2a) of Alternative Flow, if for whatever reason, selected recipe has no details in the system, should they be able to be added from the same place that the User browses recipes and recipe details? Or, from another form/location?

#### *1.3.07 Add Food Item*

<b>Use Case ID:</b>	1.3.07		
<b>Use Case Name:</b>	Add Food Item		
<b>Created By:</b>	Phillip Hansen	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-10-25	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Waiter		
<b>Description:</b>	User will be able to add food items to a guest's ticket, and can send the items back to the kitchen to be made		
<b>Trigger:</b>	User selects a food or beverage item they wish to add		
<b>Preconditions:</b>	A list of food/beverage items has been displayed to the User		
<b>Postconditions:</b>	User has added one or more food/beverage items to a guest's ticket		
<b>Normal Flow:</b>	1. User selects an item they wish to add to a ticket		

	<ol style="list-style-type: none"> <li>2. System adds the item to specified guest's ticket</li> <li>3. System displays guest's current ticket, with all items and a calculated order total</li> <li>4. System prompts User to confirm ticket items</li> <li>5. User selects confirm ticket</li> <li>6. System sends order to kitchen area</li> </ol>
<b>Alternative Flows:</b>	<p>5a. If User wants to add/remove an item to ticket</p> <ol style="list-style-type: none"> <li>1. User selects cancel</li> <li>2. Use Case resumes on step 3 of Normal Flow</li> </ol>
<b>Exceptions:</b>	<p>2a. If kitchen does not have ingredients on-hand to make/prepare item</p> <ol style="list-style-type: none"> <li>1. System alerts User that kitchen cannot make specified item at this time</li> <li>2. Use Case resumes on step 1</li> </ol> <p>6a. If there is an issue sending the order data to the kitchen</p> <ol style="list-style-type: none"> <li>1. System alerts User that there was an issue sending order data to the kitchen, prompts User to try sending again</li> <li>2. Use Case resumes on step 5 of normal Flow</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	Several times per hour, as needed.
<b>Special Requirements:</b>	Guest Name and Room Number will be displayed alongside their order information
<b>Assumptions:</b>	User is logged in  User is a Waiter, or somebody working at the restaurant needing to take down guest's orders
<b>Notes and Issues:</b>	N/A

### 1.3.10 Update Recipes

<b>Use Case ID:</b>	1.3.10		
<b>Use Case Name:</b>	Update recipes		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Food Preparer		
<b>Description:</b>	Update existing recipes in the database		
<b>Trigger:</b>	User requests system to update an existing recipe		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User must be logged in</li> <li>2. Recipe database must exist</li> <li>3. If so, recipe database must be accessible (not locked)</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. User gains access to the recipe database and can update an existing recipe</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User chooses to update an existing recipe</li> <li>2. System prompts user for the recipe they want to update</li> <li>3. If recipe exists, user is asked which information should be updated</li> <li>4. Once changes are entered, user selects submit</li> <li>5. The changes are made to the recipe in the database</li> </ol>		
<b>Alternative Flows:</b>	<p>1a. In step 1 of the normal flow, if the User is not logged in to the system</p> <ol style="list-style-type: none"> <li>1. System will prompt User to log in</li> <li>2. User enters correct login information</li> <li>3. Use Case resumes on step 2</li> </ol> <p>1b. In step 1 of the normal flow, if the User enters incorrect login information</p> <ol style="list-style-type: none"> <li>1. System will alert that login credentials were incorrect and prompt User to re-enter login information</li> <li>2. Step repeats until valid credentials are entered</li> <li>3. Use Case resumes on step 2 of normal flow</li> </ol> <p>2a. In step 2 of the normal flow, if the recipe database is not found or not accessible</p> <ol style="list-style-type: none"> <li>1. System will alert user that database was not found or is locked by someone/is unavailable</li> <li>2. Use Case terminates</li> </ol> <p>3a. In step 2 of the normal flow, if the recipe does not exist in the database</p> <ol style="list-style-type: none"> <li>1. System will alert user that the record does not exist in the database</li> <li>2. Use Case terminates</li> </ol>		
<b>Exceptions:</b>	See alternate flows (above)		

<b>Includes:</b>	View Recipes  View Recipe Details
<b>Frequency of Use:</b>	On demand, typically 1 to 50 times per day
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions
<b>Notes and Issues:</b>	N/A

#### 1.4.05 Delete Recipes

<b>Use Case ID:</b>	1.4.05		
<b>Use Case Name:</b>	Delete recipes		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Primary: Food Preparer		
<b>Description:</b>	Delete recipes in the database		
<b>Trigger:</b>	User requests to delete (deactivate) an existing recipe		
<b>Preconditions:</b>	1. User must be logged in 2. Recipe database must exist 3. If so, recipe database must be accessible (not locked)		
<b>Postconditions:</b>	1. User gains access to the recipe database and can delete an existing recipe		
<b>Normal Flow:</b>	1. User chooses to delete an existing recipe 2. System prompts user for the recipe they want to delete 3. If recipe exists, user is asked to re-confirm that they would like to delete the recipe 4. If yes, the recipe is deleted (deactivated) from the database		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the bartender is not logged in to the system  1. System will prompt user to log in 2. User enters correct login information 3. Use Case resumes on step 2  1b. In step 1 of the normal flow, if the bartender enters incorrect login information  1. System will alert that login credentials were incorrect and prompt user to re-enter login information 2. Step repeats until valid credentials are entered 3. Use Case resumes on step 2 of normal flow		

	<p>2a. In step 2 of the normal flow, if the recipe database is not found or not accessible</p> <ol style="list-style-type: none"> <li>1. System will alert user that database was not found or is locked by someone/is unavailable</li> <li>2. Use Case terminates</li> </ol> <p>3a. In step 2 of the normal flow, if the recipe does not exist in the database</p> <ol style="list-style-type: none"> <li>1. System will alert user that the record does not exist in the database</li> <li>2. Use Case terminates</li> </ol>
<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	Recipe database
<b>Frequency of Use:</b>	On demand, typically 1 to 50 times per day
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 1.4.08 Remove Dishes from Menu

<b>Use Case ID:</b>	1.4.08		
<b>Use Case Name:</b>	Remove Dishes from Menu		
<b>Created By:</b>	Austin Berquam	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	2018-10-26	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Manager		
<b>Description:</b>	Removing a dish from the menu		
<b>Trigger:</b>	Manager requests Remove Dish		
<b>Preconditions:</b>	1. A dish is in in the menu		
<b>Postconditions:</b>	1. The dish is removed from the menu.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Manager chooses Edit Dishes</li> <li>2. Manager selects dish they wish to edit</li> <li>3. Manager selects delete</li> <li>4. Dish is taken off the menu</li> </ol>		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	See alternate flows (above)		
<b>Includes:</b>	N/A		

<b>Frequency of Use:</b>	On demand, 1-365 times per year
<b>Special Requirements:</b>	1. The dish must already exist
<b>Assumptions:</b>	1. Assume that the dish already exists in the system
<b>Notes and Issues:</b>	1. Not sure if this use case is necessary.

### 2.1.02 Add Drink

<b>Use Case ID:</b>	2.1.02		
<b>Use Case Name:</b>	Add drink		
<b>Created By:</b>	Jesse Tomash	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Bartender, Drink UI		
<b>Description:</b>	Add a drink to the Database		
<b>Trigger:</b>	A new drink recipe is discovered		
<b>Preconditions:</b>	1. User has access to Drink UI 2. User has a new drink to add		
<b>Postconditions:</b>	1. A new drink recipe is in the DB		
<b>Normal Flow:</b>	1. User navigates into Drink UI 2. User selects add new drink recipe 3. System prompts for ingredients 4. User enters ingredients 5. User saves new recipe 6. Recipe is added to DB		
<b>Alternative Flows:</b>	5a. Incorrect information  1. The user is informed what information is incorrect 2. The user corrects the information and clicks save		
<b>Exceptions:</b>	4a. In step 4 of the normal flow, if we do not have carry an ingredient  1. Drink is not saved 2. User is notified of absence of ingredient		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	A couple times per month		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	User is logged in. User has permissions.		

<b>Notes and Issues:</b>	NA
--------------------------	----

### 2.1.17 Add Menu List

<b>Use Case ID:</b>	2.1.17		
<b>Use Case Name:</b>	Add Menu List		
<b>Created By:</b>	Eduardo Colon	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Admin,Menu UI(Database -Secondary)		
<b>Description:</b>	Admin wishes to add a menu option in the system, so he can provide new menu choices in the list		
<b>Trigger:</b>	1. Admin needs to add a menu option		
<b>Preconditions:</b>	1. Admin must have an active account with menu role privileges 2. Admin must be logged in		
<b>Postconditions:</b>	1. Admin will be able to add new menu option to the list		
<b>Normal Flow:</b>	1. Admin opens System settings 2. Admin selects Add a Menu Option 3. System requests a name for the new menu 4. Admin enters the name for the menu 5. System prompts Admin to confirm choice to create new Menu Option 6. Admin confirms choice 7. System displays the new created menu in the list of menus		
<b>Alternative Flows:</b>	5a. In step 5 of the Normal Flow, if Admin chooses to cancel choice 1. System resumes on step 3 of the Normal Flow.		
<b>Exceptions:</b>	N/A		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	As needed, at most 10-12 times per week.		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	User is logged in. User		
<b>Notes and Issues:</b>	There are no open issues		

### 2.2.01 Browse Menu List

<b>Use Case ID:</b>	2.2.01		
<b>Use Case Name:</b>	Browse Menu List		
<b>Created By:</b>	Eduardo Colon	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Admin ,Database (Secondary)		
<b>Description:</b>	Admin wishes to check a product in the system, so he can replenish it.		
<b>Trigger:</b>	Admin needs to browse a menu list		
<b>Preconditions:</b>	Admin must have an active account with menu role privileges Admin must be logged in		
<b>Postconditions:</b>	Admin will see the Menu List details		
<b>Normal Flow:</b>	1. Admin needs to create report in the system 2. Admin logs in to resort CMS. 3. Admin username or email. 4. Admin enters password. 5. System validates Admin. 6. Admin needs to browse a menu list 7. System displays a Menu List to choose from 8. Admin selects a Menu from the List 9. Admin press the Browse Menu button 10. System displays more details from that Menu List		
<b>Alternative Flows:</b>	1a. In step 4 of the normal flow, if the Admin are not online. 1. Admin can't request browse menu from resort or have events Admin complete the task in the system. 2. Use Case resumes on step 5		
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the Admin enters and invalid Password 1. Transaction is disapproved 2. Message to Admin to re-enter Password 3. Admin enters correct Password		

	4. Use Case resumes on step 5 of normal flow]
<b>Includes:</b>	NA
<b>Frequency of Use:</b>	On Demands: 1-50 times per day
<b>Special Requirements:</b>	Admin will need to have access to the internet to access event browse menu.
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	They have corrected permission to edit and browse menu.

#### 2.2.05 View Recipe Details

<b>Use Case ID:</b>	2.2.05		
<b>Use Case Name:</b>	View recipe details		
<b>Created By:</b>	Kevin Broskow	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Chef, Recipe Database (secondary)		
<b>Description:</b>	Allow user to view details on all recipes in the database		
<b>Trigger:</b>	User selects viewing recipe by clicking on a specific one and clicking a view details button		
<b>Preconditions:</b>	User selects viewing recipe by clicking on a specific one and clicking a view details button		
<b>Postconditions:</b>	User views a detailed window of the recipe selected		
<b>Normal Flow:</b>	1. User successfully logs into the system 2. User selects viewing specific recipes by selecting one then clicking on a view details button 3. System creates a window for viewing detailed information on recipes 4. System queries the database for the needed information 5. System populates the information fields from the database information 6. System presents user with a window that includes the detailed information		
<b>Alternative Flows:</b>	1a) If User inputs invalid login credentials		

	<ol style="list-style-type: none"> <li>1. System prompts user to re-enter login credentials</li> <li>2. User enters login credentials</li> <li>3. System validates input</li> <li>4. Use case resumes on step 2 of Normal Flow</li> </ol>
<b>Exceptions:</b>	4a) The database query fails <ol style="list-style-type: none"> <li>1.The system prints an error message to user and awaits acknowledgement</li> <li>2. Use case resumes on step 2.</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	1-5 times per hour during operating hours
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	Not certain if discontinued recipes should be included

#### 2.2.06 View Orders

<b>Use Case ID:</b>	2.2.06 View Orders		
<b>Use Case Name:</b>	Browse Events		
<b>Created By:</b>	Kevin Broskow	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-08
<b>Actors:</b>	Primary: Chef Secondary: Waiter		
<b>Description:</b>	Allow user to view current food orders		
<b>Trigger:</b>	The user selects the options to view current orders		
<b>Preconditions:</b>	The assumptions are true.		
<b>Postconditions:</b>	The users order viewing needs are satisfied		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user looks at the window to see current orders, ordered by first placed</li> <li>2. User prepares the order</li> <li>3. User marks the order as complete</li> <li>4. System removes the order from current orders</li> <li>5. System displays next upcoming order</li> </ol>		

<b>Alternative Flows:</b>	<p>1a) If there are currently no food orders in the system</p> <ol style="list-style-type: none"> <li>1. System alerts User that there are no orders currently in queue</li> <li>2. Use case resumes on step 1 of Normal Flow</li> </ol>
<b>Exceptions:</b>	<p>2a) The database query fails</p> <ol style="list-style-type: none"> <li>1. The system prints an error message and awaits acknowledgement</li> <li>2. User acknowledges the systems prompt</li> <li>3. Use case resumes on step 1 of Normal Flow</li> </ol>
<b>Includes:</b>	Create, Update, Delete Order
<b>Frequency of Use:</b>	As needed during operating hours
<b>Special Requirements:</b>	None currently.
<b>Assumptions:</b>	<p>The user is logged in.</p> <p>The user has access.</p> <p>The window for current orders is open.</p> <p>There are available orders to view.</p>
<b>Notes and Issues:</b>	In a busy kitchen, it doesn't really seem like a chef would have time to stop and mess with a program when they are done with an order... To me, it seems like the person waiting tables should maybe be the one to mark off an order as complete after the chef has fulfilled said order.

### 2.3.11 Edit Drink

<b>Use Case ID:</b>	2.3.11		
<b>Use Case Name:</b>	Edit Drink		
<b>Created By:</b>	Jesse Tomash	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Food Preparer		
<b>Description:</b>	Edit an existing Drink recipe		
<b>Trigger:</b>	A recipe is found to be wrong, or is wanted to be changed		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. A drink recipe is wrong or wanted to be changed</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Drink recipe is changed in the DB</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Food Preparer needs to create report in the system</li> <li>2. Food Preparer logs in to resort CMS.</li> </ol>		

	<ol style="list-style-type: none"> <li>3. Food Preparer username or email.</li> <li>4. Food Preparer enters password.</li> <li>5. System validates Food Preparer.</li> <li>6. Food Preparer finds a recipe to be changed</li> <li>7. Food Preparer logs into system</li> <li>8. System displays all drinks</li> <li>9. Food Preparer searches for the drink to be changed</li> <li>10. Food Preparer presses edit on the drink to be changed</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1a. In step 4 of the normal flow, if the Food Preparer are not online.             <ol style="list-style-type: none"> <li>1. Food Preparer can't request print report off from resort or have events Food Preparer complete the task in the system.</li> <li>2. Use Case resumes on step 5</li> </ol> </li> </ol>
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>3a. In step 3 of the normal flow, if the Food Preparer enters and invalid Password             <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to Food Preparer to re-enter Password</li> <li>3. Food Preparer enters correct Password</li> <li>4. Use Case resumes on step 5 of normal flow</li> </ol> </li> </ol>
<b>Includes:</b>	User Log in
<b>Frequency of Use:</b>	1 -5 times a week
<b>Special Requirements:</b>	Food Preparer will need to have access to the internet to access event Edit Drink details.
<b>Assumptions:</b>	The drink is already in the DB
<b>Notes and Issues:</b>	They have corrected permission to edit details.

#### 2.4.05 Remove Recipes

<b>Use Case ID:</b>	2.4.05		
<b>Use Case Name:</b>	Remove Recipes		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-11-07	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Food Preparer		
<b>Description:</b>	As a food prepared, I would like to deactivate recipes so that I can keep the system up to date		

<b>Trigger:</b>	User does not need the recipes any longer.
<b>Preconditions:</b>	User must be logged in Recipe database must exist If so, recipe database must be accessible (not locked)
<b>Postconditions:</b>	The recipe that has been deactivated no longer shown in the database.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User Chooses to retrieve recipes</li> <li>2. Recipe database is accessed</li> <li>3. User search recipes to the database</li> <li>4. User selects one or some recipes to be deactivated.</li> <li>5. Ask for confirmation before deactivating the data.</li> <li>6. Show message if deactivated run success.</li> </ol>
<b>Alternative Flows:</b>	<p>1a. In step 1 of the normal flow, if the User is not logged in to the system</p> <ol style="list-style-type: none"> <li>1. System will prompt User to log in</li> <li>2. User enters correct login information</li> <li>3. Use Case resumes on step 2</li> </ol> <p>1b. In step 1 of the normal flow, if the User enters incorrect login information</p> <ol style="list-style-type: none"> <li>1. System will alert that login credentials were incorrect and prompt User to re-enter login information</li> <li>2. Step repeats until valid credentials are entered</li> <li>3. Use Case resumes on step 2 of normal flow</li> </ol> <p>Note: Insert a new row for each distinctive alternative flow.</p> <p>2a. In step 2 of the normal flow, if the recipe database is not found or not accessible</p> <ol style="list-style-type: none"> <li>1. System will alert user that database was not found or is locked by someone/is unavailable</li> <li>2. Use Case terminates</li> </ol>
<b>Exceptions:</b>	6a. If the deactivated run failure, it will pop up error message and the link to send message to the administrator.
<b>Includes:</b>	Recipe database
<b>Frequency of Use:</b>	On demand, typically 1 to 50 times per month
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User know English.
<b>Notes and Issues:</b>	After deactivated, do we need to send the list deactivated recipes to user's email?

#### 2.4.08 Deactivate Menu List

<b>Use Case ID:</b>	2.4.08		
<b>Use Case Name:</b>	Deactivate Menu List		
<b>Created By:</b>	Eduardo Colon	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Admin, Database (Secondary)		
<b>Description:</b>	The Admin wishes to deactivate a menu in the system, so he can remove it, if it is no longer available		
<b>Trigger:</b>	1. System displays a Menu List tab		
<b>Preconditions:</b>	1. Admin must have an active account with menu role privileges 2. Admin must be logged in		
<b>Postconditions:</b>	what happens when the actor's goal is achieved  1. Admin will be able to deactivate a product from the Menu List details		
<b>Normal Flow:</b>	1. Admin needs to create report in the system 2. Admin logs in to resort CMS. 3. Admin username or email. 4. Admin enters password. 5. System validates Admin. 6. System displays a Menu List tab 7. Admin press the menu tab 8. System display all the menu list 9. Admin choose a Menu from the List to be deactivated 10. Admin press the deactivate Menu		
<b>Alternative Flows:</b>	9a. In step 9 of the normal flow, if there is not selected order.  1. System displays an error message 2. Admin acknowledges the error 3. The System returns to step 8 of the normal flow		
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the Admin enters and invalid Password  1. Transaction is disapproved		

	<p>2. Message to Admin to re-enter Password      3. Admin enters correct Password      4. Use Case resumes on step 5 of normal flow</p>
<b>Includes:</b>	NA
<b>Frequency of Use:</b>	1 -5 times a week
<b>Special Requirements:</b>	Admin will need to have access to the internet to access event deactivate menu
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	<p>There are no open issues      They have corrected permission to edit and deactivate menu</p>

### 3.1.05 Create Catering Request

<b>Use Case ID:</b>	3.1.05		
<b>Use Case Name:</b>	Create Catering Request		
<b>Created By:</b>	Matt LeMarche	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	2018-09-16	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary Actor: Event Manager		
<b>Description:</b>	The Event manager submits a catering request for an Event		
<b>Trigger:</b>	An Event Request was approved by the Event Manager in which the Event Host wanted the resort to Cater the event		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>An Event Manager approved an Event Request</li> <li>The Event request wanted the Resort to cater the Event</li> <li>Event Manager is logged in</li> </ol>		
<b>Postconditions:</b>	1. A form will be created and sent to the Catering department		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Prompts Manager for the Location of the Event</li> <li>Prompts Manager for the Time of the Event</li> <li>Prompts Manager for the number of people attending the Event</li> <li>Prompts the Event Manager for the menu items to cater</li> <li>Prompts Manager to enter any special requests</li> <li>A catering request is created.</li> </ol>		
<b>Alternative Flows:</b>	8a) If desired menu items are not currently in stock <ol style="list-style-type: none"> <li>System alerts Event Manager that one or more of their requested menu items are not currently in stock</li> <li>Use Case resumes on step 7</li> </ol>		
<b>Exceptions:</b>	1a) Database Error:		

	<ol style="list-style-type: none"> <li>1. Display error message</li> <li>2. Return to step 1</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	As needed
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	<p>This may be done by the system instead of manually by the Event Manager</p> <p>The Event request can be customizable based on what the event requires</p>

### 3.3.13 Update Catering Request

<b>Use Case ID:</b>	3.3.13		
<b>Use Case Name:</b>	Update Catering Request		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Alisa Roehr
<b>Date Created:</b>	2018-09-16	<b>Last Revision Date:</b>	2018-11-04
<b>Actors:</b>	Primary Actor: Event Manager		
<b>Description:</b>	The Event manager updates a submitted catering request for an Event		
<b>Trigger:</b>	An Event Manager requests to update a Catering Request for an Event		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. An Event Manager approved an Event Request</li> <li>2. The Event request wanted the Resort to cater the Event</li> <li>3. The Assumptions are true.</li> </ol>		
<b>Postconditions:</b>	1. A form will be created and sent to the Catering department		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Prompts Event Manager to choose a Catering Form to update</li> <li>2. Event Manager selects a Catering Form</li> <li>3. System returns the Catering Form</li> <li>4. Event Manager makes Changes to the Catering Form</li> <li>5. Event Manager requests to save the Catering Form</li> <li>6. System saves the updated Catering Form</li> <li>7. System Alerts Catering that a change has been made to the Catering Form</li> </ol>		
<b>Alternative Flows:</b>	None currently		
<b>Exceptions:</b>	1. The Event is too soon for Catering to accept the update		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	As needed		

<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

## Supply Operations

The following use cases detail the design relating to the order of supplies and the replenishing of inventory.

### 1.1.08 Create Supply

<b>Use Case ID:</b>	1.1.08		
<b>Use Case Name:</b>	Create Supply Order		
<b>Created By:</b>	Chase Schulte	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-09	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Manager		
<b>Description:</b>	Place orders to get more stock for inventory		
<b>Trigger:</b>	Manager initiates request for a supply order.		
<b>Preconditions:</b>	1. The Assumptions are true 2. Supplier must exist 3. Supply requested must exist		
<b>Postconditions:</b>	1. Stock will be ordered for inventory		
<b>Normal Flow:</b>	1. Manager chooses to create a supply order 2. Manager fills out all necessary fields 3. Manager submits the request 4. Supply order is sent to requested supplier		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the manager is not logged in to the system 1. System prompts manager to sign in 2. Use case resumes on step 1 1b. In step 1 of the normal flow, if the system is not available 1. System will alert user that it is not available and to try again later 2. Use Case terminates		
<b>Exceptions:</b>	See alternate flows (above)		
<b>Includes:</b>	Suppliers		
<b>Frequency of Use:</b>	Weekly		
<b>Special Requirements:</b>	Inventory must be less than max to add stock		

<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	NA

### 1.2.18 Browse Inventory

<b>Use Case ID:</b>	1.2.18		
<b>Use Case Name:</b>	Browse inventory		
<b>Created By:</b>	Chase Schulte	<b>Last Updated By:</b>	Austin Berquam
<b>Date Created:</b>	2018-09-09	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Inventory Manager(User)		
<b>Description:</b>	Browse list of inventories		
<b>Trigger:</b>	Inventory Manager requests to browse inventory		
<b>Preconditions:</b>	1. Inventory must have items		
<b>Postconditions:</b>	1. List of inventories is displayed and can be browsed		
<b>Normal Flow:</b>	1. Inventory Manager logs into to system 2. Inventory manager selects browse inventory 3. List of inventory items is displayed for browsing 4. Inventory Manager logs out		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the User is not logged in to the system 1. System will prompt User to log in 2. User enters correct login information 3. Use Case resumes on step 2  2a. In step 2 of the normal flow, if the recipe database is not found or not accessible 1. System will alert user that database was not found or is locked by someone/is unavailable 2. Use Case terminates  1b. In step 1 of the normal flow, if the User enters incorrect login information 1. System will alert that login credentials were incorrect and prompt User to re-enter login information 2. Step repeats until valid credentials are entered 3. Use Case resumes on step 2 of normal flow		

<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	System (Inventory database)
<b>Frequency of Use:</b>	On demand, 1-50 times per day
<b>Special Requirements:</b>	NA
<b>Assumptions:</b>	List of inventories has already been created. User is logged in. User has permissions.
<b>Notes and Issues:</b>	NA

#### 1.3.13 Update Quantities to Correct Mistakes

<b>Use Case ID:</b>	1.3.13		
<b>Use Case Name:</b>	Update quantities to correct mistakes		
<b>Created By:</b>	Chase Schulte	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-09	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Inventory Manager(User), Manager		
<b>Description:</b>	Place orders to get more stock for inventory		
<b>Trigger:</b>	The Customer calls the Customer Service number		
<b>Preconditions:</b>	1. Inventory must have items		
<b>Postconditions:</b>	1. Quantities are updated and fixed		
<b>Normal Flow:</b>	1. Inventory Manager logs into system 2. Inventory manager check items in inventory 3. Inventory manager finds item quantity errors and fixes them 4. Inventory Manager logs out		
<b>Alternative Flows:</b>	3a: Quantities were already correct  1. Normal flow continues at 4		
<b>Exceptions:</b>	No Items in system:  1. The System displays a message saying that there are no items in the system. 2. The Inventory Manager acknowledges this message		
<b>Includes:</b>	steps 1 and 4 are required in all flows pertaining to cataloguing inventory		
<b>Frequency of Use:</b>	Whenever an error should occur		

<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions
<b>Notes and Issues:</b>	N/A

### *1.2.17 Monitor Demand and Document Inventory*

<b>Use Case ID:</b>	1.2.17		
<b>Use Case Name:</b>	Monitor Demand and Document Inventory		
<b>Created By:</b>	Chase Schulte	<b>Last Updated By:</b>	Alisa Roehr
<b>Date Created:</b>	2018-9-9	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Inventory Manager(User)		
<b>Description:</b>	Inventory catalogues daily deliveries and shipments into database		
<b>Trigger:</b>	Inventory Manager notices demand going up/down for items		
<b>Preconditions:</b>	1. Shipments must be in the warehouse 2. The Assumptions are true		
<b>Postconditions:</b>	1. Demand will be measured, and descriptions will be added to inventory		
<b>Normal Flow:</b>	1. Inventory Manager logs on to database 2. Inventory Manager examines overall product 3. Inventory Manager determines demand based on item supplies 4. Inventory Manager catalogues demand 5. Inventory Manager logs out of database		
<b>Alternative Flows:</b>	3a. No items in inventory.  1. The Database will tell the Inventory Manager that there are no items. 2. The Inventory Manager acknowledges the message. 3. Inventory Manager logs out of database  1a. The Database is down.  1. The System will tell the Inventory Manager that a connection to the Database couldn't be made. 2. Inventory Manager acknowledges the message.		
<b>Exceptions:</b>	Inventory Manager enters wrong password.  1. Access to the Database is denied 2. Message prompts Inventory Manager to re-enter their password. 3. Inventory Manager enters correct password 4. Use Case resumes on Step 2.		
<b>Includes:</b>	Steps 1 and 5 are required in all flows pertaining to cataloguing inventory		

<b>Frequency of Use:</b>	20 times per day
<b>Special Requirements:</b>	Database must be up and running Inventory items must be present
<b>Assumptions:</b>	Inventory Manager knows how to use system. User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 1.4.07 Cancel Orders

<b>Use Case ID:</b>	1.4.07		
<b>Use Case Name:</b>	Cancel Orders		
<b>Created By:</b>	Chase Schulte	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-09	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Inventory Manager(User)		
<b>Description:</b>	Cancel an unneeded or unwanted order		
<b>Trigger:</b>	The Inventory Manager selects the option to cancel an existing order		
<b>Preconditions:</b>	1. The assumptions are true		
<b>Postconditions:</b>	1. Order is canceled		
<b>Normal Flow:</b>	1. Inventory Manager selects the order they wish to cancel 2. System returns the details of the specific order 3. Inventory manager selects the option to cancel the current order 4. The system asks the user if they are sure 5. Inventory manager confirms their decision 6. System removes order from current orders 7. The system informs the Inventory manager that their order was cancelled		
<b>Alternative Flows:</b>	5a. The inventory managers decide not to cancel their order  1. The system returns them back to step 2 of the normal flow		
<b>Exceptions:</b>	N/A		
<b>Includes:</b>	Create, Retrieve, Update Orders		
<b>Frequency of Use:</b>	As Needed		
<b>Special Requirements:</b>	N/A		

<b>Assumptions:</b>	The order they are trying to cancel is active The user is logged in The user has access
<b>Notes and Issues:</b>	N/A

#### 2.1.14 Browse Inventory System

<b>Use Case ID:</b>	2.1.14		
<b>Use Case Name:</b>	Browse inventory system		
<b>Created By:</b>	Kevin Broskow	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-19	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Manager, Inventory Database(secondary)		
<b>Description:</b>	Allow user to add items to inventory		
<b>Trigger:</b>	User has selected a button labeled add new item		
<b>Preconditions:</b>	User is logged into the inventory system  User has selected a button labeled add new item		
<b>Postconditions:</b>	A new item has been added into the inventory database		
<b>Normal Flow:</b>	1. User selects a button labeled add new item 2. System loads an empty form with all the necessary information fields 3. System awaits user entering in all pertinent information. A button labeled finish is pressed by user once information has been entered 4. System queries database to add a new record containing all the user entered information 5. System returns user to Inventory viewer		
<b>Alternative Flows:</b>	3) User has not filled in all information fields  3a) System prints an error message to user and awaits user acknowledgement  3b) System focuses on the first field that was left empty		
<b>Exceptions:</b>	4) Database query fails  4a) System prints an error message alerting user of a problem and awaits user acknowledgement  4b) System returns to basic flow step 2		

<b>Includes:</b>	Inventory System
<b>Frequency of Use:</b>	1+ Per week
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User information is spelled correctly and is accurate
<b>Notes and Issues:</b>	N/A

### 2.2.24 Search Order Supplier

<b>Use Case ID:</b>	2.2.24		
<b>Use Case Name:</b>	Search Order Supplies		
<b>Created By:</b>	Eduardo Colon	<b>Last Updated By:</b>	
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	
<b>Actors:</b>	Inventory Personnel ,Database (Secondary)		
<b>Description:</b>	The Inventory Personnel wishes to search order supplies in the system, so he can narrow his search criteria.		
<b>Trigger:</b>	1. Inventory Personnel need to search order supplies in the system		
<b>Preconditions:</b>	1. Inventory Personnel must have an active account with supply role privileges 2. Inventory Personnel must be logged in		
<b>Postconditions:</b>	1. Inventory Personnel will be able to search order supplies		
<b>Normal Flow:</b>	1. Inventory Personnel needs to create report in the system 2. Inventory Personnel logs in to resort CMS. 3. Inventory Personnel username or email. 4. Inventory Personnel enters password. 5. System validates Inventory Personnel. 6. System displays search field 7. Inventory Personnel enter the order supply to be searched 8. System returns a matching order supply record		
<b>Alternative Flows:</b>  <b>[Alternative Flow 1 Not In System]</b>	8a. In step 8 of the normal flow, if there is not matching order supply record. 1. System displays an error message 2. Inventory Personnel acknowledges the error 3. The System returns to step 6 of the normal flow.		
<b>Exceptions:</b>	In step 3 of the normal flow, if the Inventory Personnel enters an invalid Password  1. Transaction is disapproved		

	<p>2. Message to Inventory Personnel to re-enter Password</p> <p>3. Inventory Personnel enters correct Password</p> <p>4. Use Case resumes on step 5 of normal flow]</p>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	On Demand: 1 -50 times per day
<b>Special Requirements:</b>	Inventory Personnel will need to have access to the internet to access event Search Order Supply.
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	They have corrected permission to edit

#### 2.2.51 Browse Order Supply

<b>Use Case ID:</b>	2.2.51		
<b>Use Case Name:</b>	Browse Order Supply		
<b>Created By:</b>	Craig Barkley	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Manager		
<b>Description:</b>	Browse order supplies		
<b>Trigger:</b>	<p>1. Manager needs to access order to do work.</p> <p>2. Manager can prioritize work from Order view.</p>		
<b>Preconditions:</b>	<p>1. Personnel are logged in to the resort CMS</p> <p>3. Personnel has access to necessary Order views</p>		
<b>Postconditions:</b>	<p>2. Personnel have current order details</p> <p>3. Personnel has logged out of system.</p>		
<b>Normal Flow:</b>	<p>1. Personnel logs in to resort CMS.</p> <p>2. Personnel enters username or email.</p> <p>3. Personnel enters password.</p> <p>4. System validates personnel.</p> <p>5. System prompts user to select event edit view.</p> <p>6. Personnel selects current order to view.</p> <p>7. Personnel view and print details.</p> <p>8. Save updates and close records.</p>		
<b>Alternative Flows:</b>	<p>1a. In step 4 of the normal flow, if the personnel are not online.</p> <p>1. Personnel can request Events Details print off from resort or events manager.</p> <p>2. Use Case resumes on step 5</p>		

<b>Exceptions:</b>	<p>2a. In step 2 of the normal flow, if the customer enters an invalid Password</p> <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to personnel to re-enter Password</li> <li>3. Personnel enters correct Password</li> <li>4. Use Case resumes on step 4 of normal flow</li> </ol>
<b>Includes:</b>	Personnel could also email Current Orders updates. The Manager would also have access to this exact flow since they may be printing out the Event details to those who do not have access to email.
<b>Frequency of Use:</b>	This Use Case will be executed for Each Event Unless personnel have Access to online event to retrieve data themselves.
<b>Special Requirements:</b>	Personnel will need to have access to the internet to access event details.
<b>Assumptions:</b>	That Personnel will have access to the internet and that the Catering manager will follow procedure and ensure that Maintenance has accessed the event to view off details.
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. They have corrected permission to edit and save events details.</li> </ol>

#### *2.2.54 Search for Supplier Account*

<b>Use Case ID:</b>	2.2.54		
<b>Use Case Name:</b>	Search for Supplier Account		
<b>Created By:</b>	Craig Barkley	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Supplier		
<b>Description:</b>	Search Supplier account.		
<b>Trigger:</b>	<ol style="list-style-type: none"> <li>1. Manager needs to access Supplier account information</li> <li>2. Managers needs to use accessed information to complete work.</li> </ol>		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Personnel are logged in to the resort CMS</li> <li>3. Personnel has access to Supplier account records.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>2. Personnel attained new Supplier account information.</li> <li>3. Personnel Manager has logged out of system.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Personnel logs in to resort CMS.</li> <li>2. Personnel enters username or email.</li> <li>3. Personnel enters password.</li> <li>4. System validates personnel.</li> <li>5. System prompts user to select Supplier account view.</li> <li>6. Personnel retrieves data.</li> </ol>		

	7. Personnel close Supplier account records.
<b>Alternative Flows:</b>	<p>1a. In step 4 of the normal flow, if the new employees do not have access to the internet.</p> <ol style="list-style-type: none"> <li>1. Personnel can request Employee information to be printed off from resort or another manager.</li> <li>2. Use Case resumes on step 5</li> </ol>
<b>Exceptions:</b>	<p>2a. In step 2 of the normal flow, if the Supplier account enters an invalid Password</p> <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to personnel to re-enter Password</li> <li>3. Personnel enters correct Password</li> <li>4. Use Case resumes on step 4 of normal flow] Use Case resumes on step 3 of normal flow]</li> </ol>
<b>Includes:</b>	3. Personnel could also email task related information on the Supplier account. The Manager would also have access to this exact flow since they may be printing out Supplier account information details to those who do not have access to email.
<b>Frequency of Use:</b>	This Use Case will be executed to resolve Supplier account issues. Unless personnel have access to online Supplier account data themselves.
<b>Special Requirements:</b>	Personnel will need to have access to the internet to access event details.
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>1. That Personnel will have access to the internet and that the Catering manager will follow procedure and ensure that Maintenance has accessed the Supplier account details.</li> <li>2. They have corrected permission to edit and save events details.</li> </ol>
<b>Notes and Issues:</b>	There are none currently.

### 2.3.22 Edit Order Supplies

<b>Use Case ID:</b>	2.3.22		
<b>Use Case Name:</b>	Edit Order Supplies		
<b>Created By:</b>	Eduardo Colon	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Inventory Personnel, Database (Secondary)		
<b>Description:</b>	The Inventory Personnel wishes to edit order supplies in the system, so he can add or remove chosen item from it		
<b>Trigger:</b>	1. System displays an order tab		

<b>Preconditions:</b>	1. Inventory Personnel must have an active account with supply role privileges 2. Inventory Personnel must be logged in
<b>Postconditions:</b>	1. Inventory Personnel will be able to edit an order supply
<b>Normal Flow:</b>	1. System displays an order tab 2. Inventory Personnel press the order tab 3. System displays the order lists 4. Inventory Personnel select the order to be edited 5. Inventory Personnel press the edit order 6. System displays all the details for that order 7. Inventory Personnel remove or add the supplies from that order 8. System displays the new modified order with all new details
<b>Alternative Flows:</b>	4a. In step 4 of the normal flow, if there is not selected order. 1. System displays an error message 2. Inventory Personnel acknowledges the error 3. The System returns to step 3 of the normal flow
<b>Exceptions:</b>	See Alternative Flows (above)
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	On Demand.
<b>Special Requirements:</b>	Inventory personnel will need to have access to the internet to access event to edit order supplies
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	They have corrected permission to edit events order supplies.

#### 2.4.14 Cancel Order Supplies

<b>Use Case ID:</b>	2.4.14v2		
<b>Use Case Name:</b>	Cancel Order Supplies		
<b>Created By:</b>	Craig Barkley	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Manager		
<b>Description:</b>	Deactivate Order		
<b>Trigger:</b>	1. Order needs to be deactivated. 2. Saved for possible reactivation later.		
<b>Preconditions:</b>	1. Manager are logged in to the resort CMS 2. Manager has access to necessary inventories.		

<b>Postconditions:</b>	1. Manager have deactivated order. 2. Manager has logged out of system.
<b>Normal Flow:</b>	1. Manager logs in to resort CMS. 2. Manager enters username or email. 3. Manager enters password. 4. System validates Manager. 5. Manager accesses order record. 6. Manager selects to deactivate order. 7. Manager save updated record.
<b>Alternative Flows:</b>	1a. In step 4 of the normal flow, if the Manager are not online.  1. Manager can request order deactivate order off from resort or have events manager complete the task in the system. 2. Use Case resumes on step 5
<b>Exceptions:</b>	2a. In step 2 of the normal flow, if the manager enters and invalid Password  1. Transaction is disapproved 2. Message to Manager to re-enter Password 3. Manager enters correct Password 4. Use Case resumes on step 4 of normal flow
<b>Includes:</b>	2. Manager could also order deactivation updates. The Manager would also have access to this exact flow since they may be printing out the deactivation details to those who do not have access to email.
<b>Frequency of Use:</b>	This Use Case will be executed for Each order deactivation Manager has Access to online event to retrieve data and update it themselves.
<b>Special Requirements:</b>	Manager will need to have access to the internet to access event deactivation details.
<b>Assumptions:</b>	That Manager will have access to the internet and that the Catering manager will follow procedure and ensure that order deactivation is processed and that they event to view off details.
<b>Notes and Issues:</b>	1. They have corrected permission to edit and save events details.

### 3.1.16 Create New Supplier

<b>Use Case ID:</b>	3.1.16		
<b>Use Case Name:</b>	Create New Supplier		
<b>Created By:</b>	Carlos Arzu	<b>Last Updated By:</b>	Austin Berquam
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Marketing Manager or assistant		

<b>Description:</b>	The marketing department needs work that cannot be made inhouse and has suppliers that will do the work in name of the resort, arts, web advertising, database recompilation and more.
<b>Trigger:</b>	The User clicks “Add Supplier”
<b>Preconditions:</b>	1. Marketing manager or assistant are logged in.
<b>Postconditions:</b>	1. Marketing manager or assistant Save the profile.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Marketing Manager or assistant log in.</li> <li>Marketing Manager or assistant select a new profile.</li> <li>Marketing Manager or assistant fill all the requested spaces on the new profile.</li> <li>Marketing Manager or assistant check for errors.</li> <li>Marketing Manager or assistant save the profile.</li> </ol>
<b>Alternative Flows:</b>	1. Marketing Manager or assistant save, and a requested field is not filled system will return to step 3 in normal flow.
<b>Exceptions:</b>	<p>1b. In step 1 of the normal flow, if the User enters incorrect login information</p> <ol style="list-style-type: none"> <li>System will alert that login credentials were incorrect and prompt User to re-enter login information</li> <li>Step repeats until valid credentials are entered</li> <li>Once entered, Use Case resumes on step 2 of normal flow</li> </ol>
<b>Includes</b>	Supplier Information
<b>Frequency of Use:</b>	Not Frequently Used.
<b>Special Requirements:</b>	Information of the supplier is already known
<b>Assumptions:</b>	The user is logged in. The user has permissions.
<b>Notes and Issues:</b>	N/A

### 3.2.25 Browse Suppliers

<b>Use Case ID:</b>	3.2.25		
<b>Use Case Name:</b>	Browse Suppliers		
<b>Created By:</b>	Eduardo Colon	<b>Last Updated By:</b>	Eduardo Colon
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-10-30
<b>Actors:</b>	Inventory		
<b>Description:</b>	Inventory reviews inventory suppliers		
<b>Trigger:</b>	Inventory wants to see suppliers		

<b>Preconditions:</b>	Inventory must have an active account with role privileges Inventory must be logged in
<b>Postconditions:</b>	Inventory will be able to view inventory supplies
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Inventory needs to browse inventory in the system</li> <li>2. Inventory logs in to resort CMS.</li> <li>3. Inventory username or email.</li> <li>4. Inventory enters password.</li> <li>5. System validates Inventory.</li> <li>6. System displays an inventory tab</li> <li>7. Inventory presses the inventory tab</li> <li>8. System display a list of inventory suppliers</li> </ol>
<b>Alternative Flows:</b>	<p>1a. In step 4 of the normal flow, if the Inventory are not online.</p> <ol style="list-style-type: none"> <li>1. Inventory can't request browse inventory suppliers off from resort or have events Inventory complete the task in the system.</li> <li>2. Use Case resumes on step 5</li> </ol>
<b>Exceptions:</b>	<p>3a. In step 3 of the normal flow, if the Inventory enters an invalid Password</p> <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to Inventory to re-enter Password</li> <li>3. Inventory enters correct Password</li> <li>4. Use Case resumes on step 6 of normal flow]</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	1 -5 times a week
<b>Special Requirements:</b>	Inventory will need to have access to the internet to access browse Inventory suppliers.
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	They have corrected permission to browse inventories.

### 3.3.19 Update Supplier

<b>Use Case ID:</b>	3.3.19		
<b>Use Case Name:</b>	Update Supplier		
<b>Created By:</b>	Dalton Cleveland	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-11-02	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Primary Actor: Manager		
<b>Description:</b>	Edit the basic information regarding our current supplier		

<b>Trigger:</b>	User clicks on the option to update supplier
<b>Preconditions:</b>	1. The Assumptions are true.
<b>Postconditions:</b>	1. The correct supplier information is updated
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>The user selects on a specific supplier from a list of suppliers</li> <li>The system returns the details of that specific supplier</li> <li>The user enters their updated information about the supplier</li> <li>The system asks the user if they would like to save their changes</li> <li>The user confirms their changes</li> <li>The system saves, and updates the supplier information</li> <li>The system returns a message saying that the supplier was updated</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>In step 4, the user decides to cancel their changes.</li> <li>The system then returns the user to the previous screen (Details about the current supplier)</li> <li>In step 3, the user doesn't make changes and tries to submit.</li> <li>The system alerts the user that no changes were made</li> <li>Then the system returns the user back to the update screen</li> </ol>
<b>Exceptions:</b>	None currently
<b>Includes:</b>	Create, Retrieve, Delete/Deactivate Supplier
<b>Frequency of Use:</b>	As needed
<b>Special Requirements:</b>	None currently
<b>Assumptions:</b>	User is logged in. User has access. The supplier they are trying to update exists
<b>Notes and Issues:</b>	None currently

#### 4.2.17 Search Supplier Account

<b>Use Case ID:</b>	4.2.17		
<b>Use Case Name:</b>	Search Supplier Account		
<b>Created By:</b>	Dani Russo	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-11	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Primary: Employee		
<b>Description:</b>	The Employee searches a list of suppliers		
<b>Trigger:</b>	The Employee enters in search criteria and submits		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>The assumptions are true</li> <li>The Employee is viewing the list of suppliers</li> </ol>		

<b>Postconditions:</b>	1. The program displays a list of suppliers based off the search results
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The Employee enters search criteria submits</li> <li>2. The Program requests a matching record from the Database</li> <li>3. The Database returns a matching record</li> <li>4. The Program displays information inside the record</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. The Database does not find a matching record (Normal Flow, Step 3)             <ol style="list-style-type: none"> <li>a. The Program sends the Manager an error message</li> <li>b. The Employee acknowledges the message</li> <li>c. The Program returns to Normal Flow Step 1</li> </ol> </li> <li>2. In step 3 of the normal flow, if the supplier database is not found or not accessible             <ol style="list-style-type: none"> <li>1. System will alert user that database was not found or is locked by someone/is unavailable</li> <li>2. Use Case terminates</li> </ol> </li> </ol>
<b>Exceptions:</b>	N/A
<b>Includes:</b>	Supplier Database
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	None currently
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	User is logged in to the system as a member or guest.

#### 4.2.37 Access Supply Request

<b>Use Case ID:</b>	4.2.37		
<b>Use Case Name:</b>	Access Supply Requests		
<b>Created By:</b>	Dani Russo	<b>Last Updated By:</b>	Dani Russo
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Manager		
<b>Description:</b>	The Manager can view pending supply requests from other employees		
<b>Trigger:</b>	The Manager selects “New Office Orders” and submits		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The assumptions are true</li> </ol>		

<b>Postconditions:</b>	1. The Manager can view a list of all new/pending order requests.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The Manager selects “New Office Orders” and submits</li> <li>2. The Program requests a list of new orders from the database</li> <li>3. The Database returns request</li> <li>4. The Program displays a list of all new/pending order requests</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. In step 3 of the normal flow, if the order database is not found or not accessible             <ol style="list-style-type: none"> <li>1. System will alert user that database was not found or is locked by someone/is unavailable</li> <li>2. Use Case terminates</li> </ol> </li> </ol>
<b>Exceptions:</b>	N/A
<b>Includes:</b>	Order Database
<b>Frequency of Use:</b>	Daily
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 5.2.10 View Supply Order Details

<b>Use Case ID:</b>	5.2.10		
<b>Use Case Name:</b>	View Supply Order Details		
<b>Created By:</b>	Wes Richardson	<b>Last Updated By:</b>	Craig Barkley
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Manager		
<b>Description:</b>	<ol style="list-style-type: none"> <li>1. The manager reads the supply order detail and can make assessment on the supply order and its details.</li> </ol>		
<b>Trigger:</b>	<ol style="list-style-type: none"> <li>1. Details of a supply order need to be viewed to verify correctness of purchase.</li> <li>2. As well this feature can be used for planning and accounting purposes.</li> </ol>		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Navigation from home screen to supply page</li> <li>2. Click on Orders Button</li> <li>3. The order will have to be placing to see the details screen.</li> </ol>		
<b>Postconditions:</b>	1. The screen Displays the details of a supply order.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user clicks View Supply orders</li> <li>2. The user clicks the order with Supplies to view</li> <li>3. The user clicks View Detail</li> </ol>		

	4. The user can email or print supply order 5. The user closes application window.
<b>Alternative Flows:</b>	2a. In step 2 of the normal flow, if the order has not been placed the View Supply order button will not be highlighted.  1. System will prompt user of missing Detail 2. User accepts Alert 3. Use Case resumes on step 1
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the user is unable to view details  1. Request is canceled 2. Message to user unable to View Document 3. User continues to step 5.
<b>Includes:</b>	Cancel Order  Alter Order  View Supply order List
<b>Frequency of Use:</b>	Periodically as supplies are order and inventory is checked.
<b>Special Requirements:</b>	Manager needs to have access to view records.
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	Make sure order is placing to view details.

## Facility Operations

The following use cases detail the design relating to the resort maintenance and manufacturing.

### 1.2.36 Browse Hours

<b>Use Case ID:</b>	1.2.36		
<b>Use Case Name:</b>	Browse Hours		
<b>Created By:</b>	Austin Berquam	<b>Last Updated By:</b>	Dalton Cleveland, Craig Barkley
<b>Date Created:</b>	2018-10-26	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Primary: Public User		
<b>Description:</b>	A customer views the hours of the restaurant/hotel		
<b>Trigger:</b>	The actor selects to View Hours		
<b>Preconditions:</b>	1. There are hours available to view		
<b>Postconditions:</b>	1. The customer successfully views hours		

<b>Normal Flow:</b>	1. The actor selects the option to view hours 2. The system returns the list of hours for each section of the resort 3. The actor views hours
<b>Alternative Flow:</b>	1. There are no alternative flows currently
<b>Exceptions:</b>	2a. The hours have not been created  1. The systems return a statement saying that there are currently no hours posted 2. The user accepts the systems prompt 3. The system returns the user back to the home page.
<b>Includes:</b>	1. Create, Update, Delete Hours
<b>Frequency of Use:</b>	1. On demand
<b>Special Requirements:</b>	1. None currently.
<b>Assumptions:</b>	1. The user is logged in 2. The user has access
<b>Notes and Issues:</b>	1. None currently.

#### 2.1.16 Create Maintenance Ticket

<b>Use Case ID:</b>	2.1.16		
<b>Use Case Name:</b>	Create Maintenance Ticket		
<b>Created By:</b>	Kevin Broskow	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-19	<b>Last Revision Date:</b>	2018-11-08
<b>Actors:</b>	Manager, Maintenance system(secondary)		
<b>Description:</b>	Allow Manager to create a maintenance ticket		
<b>Trigger:</b>	Manager selects create new ticket button		
<b>Preconditions:</b>	The assumptions are true.		
<b>Postconditions:</b>	A maintenance ticket has been created and sent to the maintenance department		
<b>Normal Flow:</b>	1. Manager selects Create a Maintenance Ticket 2. System pulls a list from the database of the available Maintenance forms 3. Manager selects which form they would like to fill out 4. System pulls up a form containing all the necessary fields to generate a new ticket 5. Manager fills out the form 6. Manager presses submit button		

	<ol style="list-style-type: none"> <li>7. System validates form</li> <li>8. System queries database and creates a new record</li> <li>9. System prints a confirmation to Admin and awaits acknowledgement</li> <li>10. System returns user to maintenance viewer</li> </ol>
<b>Alternative Flows:</b>	<p>6a) Form has partially-filled in fields, or some fields have missing or incorrect information</p> <ol style="list-style-type: none"> <li>1. System prompts Admin to re-enter information.</li> <li>2. Use Case resumes on step 5 of Normal Flow</li> </ol>
<b>Exceptions:</b>	<p>6a) Database connection is not made or interrupted</p> <ol style="list-style-type: none"> <li>1. System alerts Admin of Database issue.</li> <li>2. Use Case resumes on step 1 of Normal Flow</li> </ol>
<b>Includes:</b>	Inventory System
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>1. All user information is entered correctly and is accurate</li> <li>2. User is logged into the maintenance system</li> <li>3. User has access</li> </ol>
<b>Notes and Issues:</b>	Assumes there will multiple types of maintenance forms with different types of data fields. Customer verbally illustrated this to Activity Diagram team.

#### 2.2.34 View Maintenance Work Orders

<b>Use Case ID:</b>	2.2.34		
<b>Use Case Name:</b>	View Maintenance Work Orders		
<b>Created By:</b>	Kevin Broskow	<b>Last Updated By:</b>	Austin Delaney
<b>Date Created:</b>	2018-09-19	<b>Last Revision Date:</b>	2018-11-08
<b>Actors:</b>	Manager, Maintenance system(secondary)		
<b>Description:</b>	Allow user to view all maintenance work orders		
<b>Trigger:</b>	User logs into the maintenance system		
<b>Preconditions:</b>	User enters a valid login		
<b>Postconditions:</b>	The user is browsing a list of current maintenance work orders		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User logs into the maintenance system</li> <li>2. System queries database to pull up all current work orders</li> <li>3. System provides a list to user</li> </ol>		

<b>Alternative Flows:</b>	NA
<b>Exceptions:</b>	<p>2) Database query fails</p> <p>2a) System prints an error message alerting user of a problem and awaits user acknowledgement</p> <p>2b) System returns to basic flow step 1</p>
<b>Includes:</b>	Maintenance System
<b>Frequency of Use:</b>	3+ Per week
<b>Special Requirements:</b>	NA
<b>Assumptions:</b>	There is an existing maintenance system
<b>Notes and Issues:</b>	Some sort of color coded status tracking?

### *3.2.32 Read Building List*

<b>Use Case ID:</b>	3.2.32		
<b>Use Case Name:</b>	Read Building List		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Austin Berquam
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Event Host		
<b>Description:</b>	An Event Host wants to view a List of Buildings and the schedules for them		
<b>Trigger:</b>	Event Host requests to view a List of Buildings and the schedules for the		
<b>Preconditions:</b>	1. The assumptions are true		
<b>Postconditions:</b>	1. The Event Host will be able to view the list of all buildings available for reservations		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Event Host requests to see the list of all Buildings</li> <li>2. System returns all Buildings which are available to reserve</li> <li>3. Event Host selects a Building</li> <li>4. System returns more information on the Building including schedule</li> </ol>		
<b>Alternative Flows:</b>	<p>2a) If database query fails</p> <ol style="list-style-type: none"> <li>1. System alerts Event Host, prompts them to try their search again</li> <li>2. Use Case resumes on step 1 of Normal Flow</li> </ol>		
<b>Exceptions:</b>	<p>2a) If database query fails</p> <ol style="list-style-type: none"> <li>3. System alerts Event Host, prompts them to try their search again</li> </ol>		

	4. Use Case resumes on step 1 of Normal Flow
<b>Includes:</b>	The list of reservations
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	Reservations and available rooms schedule has been made. User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

### *5.1.02 Add Reservation*

<b>Use Case ID:</b>	5.1.02		
<b>Use Case Name:</b>	Add Reservation		
<b>Created By:</b>	Wes Richardson	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Receptionists or Concierge		
<b>Description:</b>	The user will add a reservation for the Resort Customer		
<b>Trigger:</b>	A Resort Customer calls to make a reservation		
<b>Preconditions:</b>	1. User is an employee of Resort 2. The user has the role of Receptionist or Concierge		
<b>Postconditions:</b>	1. The user is informed that the reservation was made and is returned to the starting screen		
<b>Normal Flow:</b>	1. The user clicks the Button "Make Reservation" 2. The data entry screen comes up and the user fills in the text boxes 3. The user clicks the submit button 4. The system tells the user "Reservation complete"		
<b>Alternative Flows:</b>	3a. Information is missing 1. The system will inform the user to fill out all boxes 3b. Incorrect information (Letter is a number field) 1. The system will inform the user what information is incorrect		
<b>Exceptions:</b>	3c. Database Issues 1. The system informs the user of the database issue and that it did not save the reservation		
<b>Includes:</b>	System		

<b>Frequency of Use:</b>	3 per hour
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 5.2.01 View Details of Reservation

<b>Use Case ID:</b>	5.2.01		
<b>Use Case Name:</b>	View Details of Reservation		
<b>Created By:</b>	Wes Richardson	<b>Last Updated By:</b>	Craig Barkley
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	receptionists, Managers, Guest		
<b>Description:</b>	<p>1. Receptionists read the reservations detail and can make assessment on the reservations and its details.</p>		
<b>Trigger:</b>	<p>1. Details of a reservations need to be viewed to verify correctness of purchase.</p> <p>2. As well This feature can be used for booking, planning and accounting purposes.</p>		
<b>Preconditions:</b>	<p>1. Navigation from home screen to reservations page</p> <p>2. Click on reservations Button</p> <p>3. The reservations will have to be place in reservations to see the details screen.</p>		
<b>Postconditions:</b>	1. The screen Displays the details of a reservations.		
<b>Normal Flow:</b>	<p>1. The user clicks View reservations</p> <p>2. The user clicks the reservations with reservations to view</p> <p>3. The user clicks View Detail</p> <p>4. The user can email or print reservations</p> <p>5. The user closes application window.</p>		
<b>Alternative Flows:</b>	<p>2a. In step 2 of the normal flow, if the reservations have not been placed the View reservations button will not be highlighted.</p> <p>1. System will prompt user of missing Detail</p> <p>2. User accepts Alert</p> <p>3. Use Case resumes on step 1</p>		
<b>Exceptions:</b>	<p>3a. In step 3 of the normal flow, if the user is unable to view details</p> <p>1. Request is canceled</p> <p>2. Message to user unable to View Document</p> <p>3. User continues to step 5.</p>		

<b>Includes:</b>	Cancel reservations Alter reservations View reservations List
<b>Frequency of Use:</b>	Periodically as reservations are checked.
<b>Special Requirements:</b>	receptionists need to have access to view records.
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	Make sure reservations is place in reservations to view details.

#### *5.2.06 View List of Reservations*

<b>Use Case ID:</b>	5.2.06		
<b>Use Case Name:</b>	View List of Reservations		
<b>Created By:</b>	Wes Richardson	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary Actor: Receptionist		
<b>Description:</b>	The screen will display a list of reservations		
<b>Trigger:</b>	The user selects the option to view current reservations		
<b>Preconditions:</b>	The assumptions are true.		
<b>Postconditions:</b>	The users' reservation viewing needs are satisfied.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user selects the option to view reservations</li> <li>2. The system returns the list of current reservations</li> <li>3. The user reads through the list of reservations</li> <li>4. The user is satisfied.</li> <li>5. The user closes the window.</li> </ol>		
<b>Alternative Flows:</b>	<p>2a. There are no reservations to view</p> <ol style="list-style-type: none"> <li>1. The system alerts the user that there are currently no reservations for it to display</li> <li>2. User accepts the systems prompt.</li> <li>3. System closes the window and returns user to previous page.</li> </ol>		
<b>Exceptions:</b>	<p>2a. The system has an error loading the list of reservations</p> <ol style="list-style-type: none"> <li>1. The system alerts the user that it cannot retrieve the list of current reservations</li> <li>2. The user accepts the prompt from the system</li> <li>3. System closes the window and returns the user to the previous page</li> </ol>		

<b>Includes:</b>	Create, Update, Delete Reservation
<b>Frequency of Use:</b>	On demand.
<b>Special Requirements:</b>	None currently.
<b>Assumptions:</b>	The user is logged in. The user has access.
<b>Notes and Issues:</b>	None currently.

#### 5.2.08 View Maintenance Ticket Details

<b>Use Case ID:</b>	5.2.08		
<b>Use Case Name:</b>	View Maintenance Ticket Details		
<b>Created By:</b>	Wes Richardson	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	9-14-2018-09-14	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Receptionists or Concierge Maintenance		
<b>Description:</b>	View Maintenance Ticket Details		
<b>Trigger:</b>	The user clicks view details on an item in the list of Maintenance Tickets		
<b>Preconditions:</b>	The assumptions are true		
<b>Postconditions:</b>	The screen displays the details of a maintenance ticket		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user clicks view maintenance tickets</li> <li>2. The system displays the list of current maintenance tickets</li> <li>3. The user clicks view details button next to the list item</li> <li>4. The system displays the details of the maintenance ticket</li> <li>5. The user reads the details of the ticket</li> <li>6. The user closes the detail window</li> <li>7. The system returns the user to the previous screen</li> </ol>		
<b>Alternative Flows:</b>	<p>1a. The system cannot find the list of maintenance tickets</p> <ol style="list-style-type: none"> <li>1. The system alerts the user that it could not find any current tickets</li> <li>2. The user acknowledges the systems prompt</li> <li>3. The system returns the user back to the previous page</li> </ol>		
<b>Exceptions:</b>	Database issues		
<b>Includes:</b>	View list of Maintenance Tickets Alter Maintenance Ticket		

	Mark Maintenance Ticket as Complete
<b>Frequency of Use:</b>	As needed
<b>Special Requirements:</b>	None currently
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	None currently

#### 5.2.09 View Room Details

<b>Use Case ID:</b>	5.2.09		
<b>Use Case Name:</b>	View Room Details		
<b>Created By:</b>	Wes Richardson	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Receptionists, Manager, Guest		
<b>Description:</b>	View Room Details		
<b>Trigger:</b>	The user needs to look up the details of a certain room		
<b>Preconditions:</b>	Receptionist already in the system.		
<b>Postconditions:</b>	The details of the selected room are displayed on the screen		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user clicks view Rooms</li> <li>2. The user clicks the room they wish to view</li> <li>3. The user clicks details</li> </ol>		
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	1a. If the online link doesn't work, it will pop the link to the offline link		
<b>Exceptions:</b>	1b. Database issues <ol style="list-style-type: none"> <li>1. The user is informed what the database issue is</li> <li>2. The user is informed cannot display information</li> </ol>		
<b>Includes:</b>	<ol style="list-style-type: none"> <li>1. View Rooms list</li> <li>2. Alter Room Status</li> </ol>		
<b>Frequency of Use:</b>	200 times a day		

<b>Special Requirements:</b>	NA
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	NA

### [5.3.15 Update Maintenance Ticket](#)

<b>Use Case ID:</b>	5.3.15		
<b>Use Case Name:</b>	Update Maintenance Ticket		
<b>Created By:</b>	James Heim	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-08
<b>Actors:</b>	Shuttle Driver		
<b>Description:</b>	To update the progress of a maintenance ticket.		
<b>Trigger:</b>	User selects the option to update maintenance ticket		
<b>Preconditions:</b>	1. The assumptions are true 2. A ticket has already been made.		
<b>Postconditions:</b>	1. Ticket is updated		
<b>Normal Flow:</b>	1. User selects a specific maintenance ticket 2. System returns the details about that ticket 3. User selects the option to update ticket 4. System allows user to make changes to the ticket 5. User makes their desired changes to the ticket. 6. User chooses the option to save changes 7. System asks the user if they are sure they would like to submit their changes 8. System saves the users changes and shows the tickets updated details		
<b>Alternative Flows:</b>	6a. The system asks the user if they would like to save their changes <ol style="list-style-type: none"> <li>The user accepts the offer to save changes</li> <li>The use case resumes on step 8 of the normal flow</li> </ol> 6b. The system asks the user if they would like to save their changes <ol style="list-style-type: none"> <li>The user declines the offer to save changes</li> <li>The use case resumes on step 4 of the normal flow</li> </ol>		
<b>Exceptions:</b>	6a. The user doesn't make any changes and tries to save. <ol style="list-style-type: none"> <li>The system alerts the user that no changes were made.</li> <li>Returns user to step 4 of the normal flow.</li> </ol>		

<b>Includes:</b>	Create, Retrieve, Delete Maintenance Ticket
<b>Frequency of Use:</b>	As needed.
<b>Special Requirements:</b>	None currently.
<b>Assumptions:</b>	User is logged in User has access
<b>Notes and Issues:</b>	None currently.

#### *5.4.03 Deactivate Open Maintenance Slip*

<b>Use Case ID:</b>	5.4.03		
<b>Use Case Name:</b>	Deactivate Open Maintenance Slip		
<b>Created By:</b>	Dalton Cleveland	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-10-31
<b>Actors:</b>	Maintenance Worker		
<b>Description:</b>	Closes an open maintenance slip that has been completed		
<b>Trigger:</b>	Worker clicks on the option to close a slip		
<b>Preconditions:</b>	1. The Assumptions are true		
<b>Postconditions:</b>	1. The maintenance slip is closed.		
<b>Normal Flow:</b>  <b>(The Normal Steps that would happen if all proceeded correctly)</b>	1. The user selects the option to view a specific maintenance slip 2. The system returns the details of the corresponding slip 3. The user selects the option to deactivate the slip 4. The system removes the slip from the list of open slips 5. The system adds the slip to the list of closed slips		
<b>Alternative Flows:</b>	1. There are no alternate flows		
<b>Exceptions:</b>	There are no exceptions		
<b>Includes:</b>	Create, Retrieve, Update Maintenance Slips		
<b>Frequency of Use:</b>	As Needed (Whenever A Slip Is Completed)		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	User is logged in. User has permissions.		
<b>Notes and Issues:</b>	1. NA		

#### 5.4.06 Deactivate Room

<b>Use Case ID:</b>	5.4.06		
<b>Use Case Name:</b>	Deactivate Room		
<b>Created By:</b>	Matthew Hill	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Janitor  System.		
<b>Description:</b>	Report a room's status as "not in service".		
<b>Trigger:</b>	User opens employee portal.		
<b>Preconditions:</b>	1. User needs to be logged out. 2. System needs to be online. 3. Room needs to be decided by manager as not currently habitable.		
<b>Postconditions:</b>	1. User needs to be logged out.		
<b>Normal Flow:</b>	1. User opens employee portal. 2. User inputs employee ID. 3. User inputs password. 4. User submits input. 5. System validates user's input. 6. System prompts user for desired operation. 7. User selects "report room as not in service" option. 8. System prompts User for room number. 9. User inputs room number. 10. System validates input. 11. System asks User to confirm their choice. 12. User confirms their choice. 13. System acknowledges User's decision. 14. User logs out of the system.		
<b>Alternative Flows:</b>  <b>[Alternative Flow 1 – Not in Network]</b>	Step 5: If user's log-in input is invalid, 1. System prompts user to re-enter valid data. 2. User submits input. 3. System validates user's input. 4. Input is valid, use case resumes on step 6. Step 10: If room number is invalid, 1. System prompts user to re-enter valid room number. 2. User submits input. 3. System validates input. 4. Input is valid, use case resumes on step 11.		

	Step 11: If user decides to not confirm, 1. System acknowledges user's decision. 2. Use case resumes on step 8.
<b>Exceptions:</b>	N/A
<b>Includes:</b>	Steps 1-6 are included with all use cases involving use of the employee portal.
<b>Frequency of Use:</b>	1-10 times per day.
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	1. User is an employee of the resort hotel. 2. User is a maid / janitor of the resort hotel.
<b>Notes and Issues:</b>	N/A

## Guest Operations

The following use cases detail the design relating to the guest and member services including reservations and check in/out processes.

### 1.3.20 Checkout

<b>Use Case ID:</b>	1.3.20		
<b>Use Case Name:</b>	Checkout		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Primary Actor: Front Desk Clerk/Receptionist		
<b>Description:</b>	Guest wishes to check out of the Resort		
<b>Trigger:</b>	User selects the option to checkout of resort		
<b>Preconditions:</b>	1. The assumptions are true		
<b>Postconditions:</b>	1. Guest has paid for their tab 2. The Room the guest stayed in is updated to vacant		
<b>Normal Flow:</b>	1. Primary Actor requests the Guest information 2. System returns Guest Information 3. Primary Actor requests to check the Guest out of the Resort 4. System performs Checkout actions		
<b>Alternative Flows:</b>	2a. The Guest's information isn't in the system. 3. The System shows a message saying that the Guest's information wasn't		

	<p>found.</p> <p>4. Primary Actor acknowledges the message.</p> <p>5. The program returns to step 1.</p>
<b>Exceptions:</b>	<p>1a. Database issues</p> <ol style="list-style-type: none"> <li>1. The user is informed of the database issues</li> <li>2. The user is returned to the starting screen</li> </ol>
<b>Includes:</b>	Check-In
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	None currently
<b>Assumptions:</b>	<p>The Guest is currently staying at our Resort</p> <p>The user is logged in</p> <p>The user has access</p>
<b>Notes and Issues:</b>	None currently

### 3.4.11 Delete Account

<b>Use Case ID:</b>	3.4.11		
<b>Use Case Name:</b>	Delete Account		
<b>Created By:</b>	Jacob Miller	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Web Visitor		
<b>Description:</b>	A Web Visitor has deactivated their account		
<b>Trigger:</b>	A Web Visitor requests to leave their schedule		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The assumptions are true</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The Web Visitor has deactivated their account</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The Web Visitor requests to remove their account</li> <li>2. The system asks if they are sure</li> <li>3. The Web Visitor confirms</li> <li>4. The system deactivates the account</li> </ol>		
<b>Alternative Flows:</b>	<p>3a. In Step 3 of the normal flow if the Web Visitor declines</p> <ol style="list-style-type: none"> <li>1. The system returns them to the previous screen.</li> </ol>		
<b>[Alternative Flow 1 – Not in Network]</b>			
<b>Exceptions:</b>	Not sure currently		

<b>Includes:</b>	Create, Retrieve, Update Account
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	Not sure currently
<b>Assumptions:</b>	The user is having an existing account The user is logged in The user has access
<b>Notes and Issues:</b>	None

### 5.5.01 Guest Check In

<b>Use Case ID:</b>	5.5.01		
<b>Use Case Name:</b>	Guest Check IN		
<b>Created By:</b>	Craig Barkley	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-10-31	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Receptionist		
<b>Description:</b>	Create new Guest Schedule		
<b>Trigger:</b>	1. A new guest has arrived, and a room needs to be booked. 2. Managers needs to schedule within the system.		
<b>Preconditions:</b>	1. Receptionist are logged in to the resort CMS. 2. Receptionist have access to enter in customer data and assign room.		
<b>Postconditions:</b>	1. Receptionist has created a guest account. 2. Receptionist has logged out of system.		
<b>Normal Flow:</b>	1. The Check-In Screen, choose between. 2. Repeat Guests. 3. Identifying VIPs. 4. View Guest or enter in new guest information. 5. Room Number assigned to guest. 6. Available Credit Line is opened. 7. Deposit Payment can be accepted at this time. 8. Review this Reservation 9. Answer any questions.		
<b>Alternative Flows:</b>	1a. In step 4 of the normal flow, if the new Guests do not have access to the internet.  1. Receptionist can request Guest information to be print off from resort or another manager.		

	2. Use Case resumes on step 5
<b>Exceptions:</b>	<p>2a. In step 6 of the normal flow, if the customer enters an invalid Payment</p> <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to Receptionist to re-enter information</li> <li>3. Receptionist enters correct Credit Card.</li> <li>4. Use Case resumes on step 4 of normal flow</li> </ol>
<b>Includes:</b>	Receptionist could also email updated guest information. The Manager would also have access to this exact flow since they may be printing out Guest information details to those who do not have access to email.
<b>Frequency of Use:</b>	This Use Case will be executed for Guests Unless Receptionist has Access to online view and request schedules themselves.
<b>Special Requirements:</b>	Receptionist will need to have access to the internet to access schedule details.
<b>Assumptions:</b>	That Receptionist will have access to the internet and that the Catering manager will follow procedure and ensure that Guests have accessed the schedule details.
<b>Notes and Issues:</b>	1. They have corrected permission to edit and save events details.

### 5.3.06 Edit Room Reservation

<b>Use Case ID:</b>	5.3.06		
<b>Use Case Name:</b>	Edit Room Reservation		
<b>Created By:</b>	Wes Richardson	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	9-14-2018	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Receptionist, Manager, Guest		
<b>Description:</b>	To change some aspect of a Reservation for a Resort customer		
<b>Trigger:</b>	The Resort customer requests a change or an error was made		
<b>Preconditions:</b>	The user is logged in		
<b>Postconditions:</b>	The selected reservation has been altered and the user is returned to the starting screen		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user brings up a list of reservations</li> <li>2. The user selects the reservation they wish to alter</li> <li>3. The user clicks details</li> <li>4. The user selects edit reservation</li> <li>5. The user alters the parts of the reservation they wish to alter</li> </ol>		

	<p>6. The user clicks submit      7. The system informs the user that the reservation has been saved      8. The user is returned to the starting screen</p>
<b>Alternative Flows:</b>	<p>6a.In step 6, The Reservation was not updated</p> <ol style="list-style-type: none"> <li>1. The user is informed Reservation not updated</li> <li>2. The user hits “OK”</li> <li>3. The user is returned to the update reservation list</li> </ol>
<b>Exceptions:</b>	<p>6a. The user doesn't have all the needed boxes filled out, the system doesn't let the user continue.</p> <p>6b.The user has filled out incorrect information the system doesn't let the user continue i.e. non-numbers in a date field</p>
<b>Includes:</b>	<p>View Reservation List      Remove Reservation      View Reservation Details</p>
<b>Frequency of Use:</b>	3 times a day
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 5.4.02 Remove Reservation

<b>Use Case ID:</b>	5.4.02		
<b>Use Case Name:</b>	Remove Reservation		
<b>Created By:</b>	Wes Richardson	<b>Last Updated By:</b>	Gunardi Saputra  Wes Richardson
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-02  2018-11-09
<b>Actors:</b>	Receptionist, Manager, Guest		
<b>Description:</b>	As a receptionist, I would like to Deactivate a Reservation so that I can delete the reservation		
<b>Trigger:</b>	Resort customer wants to cancel Reservation		

<b>Preconditions:</b>	1. User is an employee of Resort 2. The user has the role of Receptionist or Concierge
<b>Postconditions:</b>	1. The user successfully removes reservation and is informed 2. The reservation was not removed, and the user is told why
<b>Normal Flow:</b>	1. The user looks up the reservations from a list 2. The user clicks to look at the details of a reservation 3. The user clicks the remove reservation button 4. The user is asked to confirm they want to remove the reservation 5. The user is returned to the starting screen
<b>Alternative Flows:</b>	2.a. If the user cannot find the reservations from a list, it will pop up an error message.  3.a. If the user cannot see the detail of the reservation, it can continue to normal flow step
<b>Exceptions:</b>	1a. Database Issues  1. The user is informed of the database issue 2. The user is returned to the starting screen
<b>Includes:</b>	1. View List of Reservation 2. View Details of Reservation
<b>Frequency of Use:</b>	50 per week
<b>Special Requirements:</b>	NA
<b>Assumptions:</b>	The user is logged in. The user has permissions
<b>Notes and Issues:</b>	1. Do we need to save the history for any removing reservation?

#### 6.1.06 Create User Account

Use Case ID:	6.1.06		
Use Case Name:	Create User Account		
Created By:	Gunardi Saputra	Last Updated By:	Gunardi Saputra
Date Created:	2018-09-20	Last Revision Date:	2018-11-05
Actors:	Primary: Guest		
Description:	Create new user account so I can use the system with my account.		
Trigger:	Guest want to use child care system to.		
Preconditions:	1. Guest has active email address for email confirmation 2. Guest has active US phone number		

Postconditions:	<ol style="list-style-type: none"> <li>1. Guest has a user login</li> <li>2. Guest need to click confirmation email to activate the new user account</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. Guest enter user login (email address)</li> <li>2. System validates if Guest 's email is already in the database</li> <li>3. Guest enters password</li> <li>4. Guest reenter password</li> <li>5. System prompts user to check email and make a validation.</li> <li>6. Guest validates email and click the link to sign in</li> </ol>
Alternative Flows:	<p>2a. In step 2 of the normal flow, if the Guest is already in the database</p> <ol style="list-style-type: none"> <li>1. System will prompt Guest to enter the password</li> <li>2. Log in</li> </ol> <p>2b. In step 2 of the normal flow, if the customer is not in the bank network</p> <ol style="list-style-type: none"> <li>1. System will prompt Guest to create user account</li> <li>2. Guest enter user login (email address)</li> <li>3. System will check if Guest's email is already in the database</li> <li>4. Use Case resumes on step 3 of normal flow.</li> </ol>
Exceptions:	<p>1a. In step 1 of the normal flow, if the Guest enters invalid email address</p> <ol style="list-style-type: none"> <li>1. System will prompt Guest to enter the right email format</li> <li>2. Message to Guest correct email address</li> </ol> <p>3a. In step 3 of the normal flow, if the Guest enters invalid password pattern or less than minimum requirement</p> <ol style="list-style-type: none"> <li>1. System will prompt Guest to reenter the right password</li> <li>2. Message to Guest correct password</li> </ol> <p>4a. In step 4 of the normal flow, if the Guest enters different password confirmation</p> <ol style="list-style-type: none"> <li>1. System will prompt Guest to reenter the confirmation password</li> <li>2. Message to Guest that the confirmation password is different with the first password.</li> </ol>
Includes:	n/a
Frequency of Use:	50 per day
Special Requirements:	n/a
Assumptions:	Guest understand English and follow the message instruction. User is logged in. User has permissions.

Notes and Issues:	1. What is the minimum password character? 2. What is the maximum password character?
-------------------	--

#### *6.4.04 Deactivate Guest Account*

<b>Use Case ID:</b>	6.4.04		
<b>Use Case Name:</b>	Deactivate Guest account		
<b>Created By:</b>	Gunardi Saputra	<b>Last Updated By:</b>	Matthew Hill, Gunardi Saputra
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-10-26, 2018-10-31
<b>Actors:</b>	Primary: Manager		
<b>Description:</b>	A Manager wishes to deactivate an account due to its inactivity		
<b>Trigger:</b>	Manager wants to view a list of currently active accounts		
<b>Preconditions:</b>	1. Manager is logged in		
<b>Postconditions:</b>	1. Manager can deactivate any account due to inactivity		
<b>Normal Flow:</b>	1. Manager logs into the system 2. System validates login credentials 3. Manager selects Guests menu 4. Select Account 5. Select Deactivate Account 6. System displays a list of accounts. 7. Manager selects an account to deactivate 8. System prompts for confirmation to deactivate 9. Manager confirms choice. 10. Account is deactivated		
<b>Alternative Flows:</b>  <b>[Alternative Flow 1 – Not in Network]</b>	2a. If Manager inputs invalid login credentials 1. System informs Manager of error and prompts them to re-enter login credentials 2. Manager re-enters login credentials. 3. Use Case resumes on step 2 of the Normal Flow 9a. If Manager cancels choice 1. System confirms choice. 2. Use Case resumes on step 6 of the Normal Flow		

<b>Exceptions:</b>	2a. If Manager inputs invalid login credentials 4. System informs Manager of error and prompts them to re-enter login credentials 5. Manager re-enters login credentials. 6. Use Case resumes on step 2 of the Normal Flow
<b>Includes:</b>	Save the update to the database.
<b>Frequency of Use:</b>	50 per month
<b>Special Requirements:</b>	Call or contact inactive guest before deactivating.
<b>Assumptions:</b>	User is an actual Manager and is logged into the system.
<b>Notes and Issues:</b>	Do we need to send the inactive guest list to the manager 2 week before deactivation?

## Personnel Management

The following use cases detail the design relating to the employees and their roles and schedule management and member services including reservations and check in/out processes, as well as general logging in and out.

### 1.2.01 Log In

<b>Use Case ID:</b>	1.2.01		
<b>Use Case Name:</b>	Log in		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-09-10	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	User		
<b>Description:</b>	User wishes to log in and access features available to their specific role		
<b>Trigger:</b>	User enters user name and submits		
<b>Preconditions:</b>	1. The Assumptions are true		
<b>Postconditions:</b>	1. User is logged in 2. The program has provided access to the User's functionality		
<b>Normal Flow:</b>	1. The User enters their username 2. The Program requests a password 3. The User enters their password and submits 4. The Program requests from the Data Store a User record matching the username and password 5. The Data Store returns a matching User record		

	6. The Program displays the User's functionality options
<b>Alternative Flows:</b>	5a. If there is no matching User Record <ul style="list-style-type: none"> <li>1. The Program displays an error message</li> <li>2. The User acknowledges the error</li> <li>a. The Program returns to Basic Flow Step 1</li> </ul>
<b>Exceptions:</b>	3a. If the User inputs invalid login credentials <ul style="list-style-type: none"> <li>1. The system alerts the User that invalid login credentials were given, and prompts the User to re-enter login credentials</li> <li>2. The User submits valid login credentials</li> <li>3. The System validates login credentials</li> <li>4. Use Case resumes on step 4 of Normal Flow</li> </ul>
<b>Includes:</b>	Save the log in history for every user.
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user interface is in the area's native language.
<b>Notes and Issues:</b>	1. How many failed attempts can a user?

### 1.2.02 Log Out

<b>Use Case ID:</b>	1.2.02		
<b>Use Case Name:</b>	Log Out		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Gunardi Saputra
<b>Date Created:</b>	2018-09-10	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	User		
<b>Description:</b>	The User wishes to log out of the system		
<b>Trigger:</b>	The User selects Log out Option on the interface		
<b>Preconditions:</b>	1. The Assumptions are true		
<b>Postconditions:</b>	1. The User is no longer logged in.		
<b>Normal Flow:</b>	1. The User Selects the log out Option on the interface 2. The Program requests confirmation for logout request 3. User confirms request 4. The Program logs the User out and goes to 1.2.01 step one		
<b>Alternative Flows:</b>	1. The User declines the confirmation to log out (Normal Flow step 3) <ul style="list-style-type: none"> <li>a. The Program exits basic flow</li> </ul>		
<b>Exceptions:</b>	1. N/A		

<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The interface is in the area's native language  User is logged in.
<b>Notes and Issues:</b>	1. Do we need to terminate the connection after some period not active in the system?

#### 1.1.19 Create New Employee Role

<b>Use Case ID:</b>	1.1.19		
<b>Use Case Name:</b>	Create new employee role		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Alisa Roehr
<b>Date Created:</b>	2018-09-21	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Primary: Manager		
<b>Description:</b>	Create new employee role		
<b>Trigger:</b>	Manager requests new employee role to be created		
<b>Preconditions:</b>	1. The Assumptions are true		
<b>Postconditions:</b>	1. If logged in and request is submitted successfully, new employee role will be added to the system		
<b>Normal Flow:</b>	1. Manager chooses to create new employee role 2. Manager fills in the new role details 3. Manager completes the request 4. New employee role is created in the system		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the Manager is not logged in to the system  1. System prompts the Manager to log in 2. Once correct credentials are provided and submitted, Manager will be logged in 3. Use Case resumes on step 1  1b. In step 1 of the normal flow, if the system is not available  1. System will alert user that it is not available and to try again later 2. Use Case terminates		

<b>Exceptions:</b>	3a. If the role submitted already exists <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to User that the role already exists</li> <li>3. User corrects the submission by entering non-duplicate information</li> <li>4. Use Case resumes on step 4</li> </ol>
<b>Includes:</b>	System
<b>Frequency of Use:</b>	On demand, 1-50 times per week
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 1.1.20 Create New Employee Schedule

<b>Use Case ID:</b>	1.1.20		
<b>Use Case Name:</b>	Create new employee schedule		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Primary: Supervisor		
<b>Description:</b>	Create new employee schedule		
<b>Trigger:</b>	User requests new employee schedule to be created		
<b>Preconditions:</b>	1. User must be logged in 2. Selected employee profile must be active (have at least one role)		
<b>Postconditions:</b>	1. If logged in and request is submitted successfully, and employee profile is active, new employee schedule will be added to the system		
<b>Normal Flow:</b>	1. User chooses to create new employee schedule 2. User chooses employee to create the schedule for 3. User fills in the new schedule 4. User completes the request 5. New employee schedule is created in the system		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the User is not logged in to the system <ol style="list-style-type: none"> <li>1. System prompts the User to sign in</li> <li>2. Use Case resumes on step 1</li> </ol>		

	<p>1b. In step 1 of the normal flow, if the system is not available</p> <ol style="list-style-type: none"> <li>1. System will alert user that it is not available and to try again later</li> <li>2. Use Case terminates</li> </ol> <p>2a. In step 2 of the normal flow, if the employee is not active</p> <ol style="list-style-type: none"> <li>1. System will alert user that employee is not active</li> <li>2. Use Case terminates</li> </ol>
<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	System
<b>Frequency of Use:</b>	On demand, 1-50 times per week
<b>Special Requirements:</b>	Only users with permission can create employee schedules.
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 1.2.45 Retrieve Employee Roles

<b>Use Case ID:</b>	1.2.45		
<b>Use Case Name:</b>	Retrieve employee roles		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Supervisor		
<b>Description:</b>	Retrieve employee roles		
<b>Trigger:</b>	User requests to access employee roles		
<b>Preconditions:</b>	1. User must be logged in		
<b>Postconditions:</b>	1. If logged in and request is submitted successfully, employee roles are retrieved and available for browsing		
<b>Normal Flow:</b>	1. User chooses retrieve employee roles 2. Employee roles are fetched from the system (database) 3. User can choose to browse the records		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the User is not logged in to the system <ol style="list-style-type: none"> <li>1. System prompts the User to sign in</li> <li>2. Use Case resumes on step 1</li> </ol> 1b. In step 1 of the normal flow, if the system is not available <ol style="list-style-type: none"> <li>1. System will alert user that it is not available and to try again later</li> <li>2. Use Case terminates</li> </ol>		

<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	View Employee Roles
<b>Frequency of Use:</b>	On demand, 1-100 times per year
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions
<b>Notes and Issues:</b>	N/A

#### 1.2.49 Retrieve Employee Schedule

<b>Use Case ID:</b>	1.2.49		
<b>Use Case Name:</b>	Retrieve employee schedule		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Primary: Supervisor		
<b>Description:</b>	Retrieve selected employee schedule from browsed list		
<b>Trigger:</b>	User requests to retrieve an employee schedule		
<b>Preconditions:</b>	1. User must be logged in 2. Employee schedule must have already been created		
<b>Postconditions:</b>	1. If logged in and employee schedule was already created, and request is submitted successfully, employee schedule will be retrieved		
<b>Normal Flow:</b>	1. User chooses an employee schedule from the browsed list 2. User chooses view schedule		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the User is not logged in to the system <ol style="list-style-type: none"> <li>System prompts the User to sign in</li> <li>Use Case resumes on step 1</li> </ol> 1b. In step 1 of the normal flow, if the system is not available <ol style="list-style-type: none"> <li>System will alert user that it is not available and to try again later</li> <li>Use Case terminates</li> </ol> 2a. In step 2 of the normal flow, if the employee schedule does not exist <ol style="list-style-type: none"> <li>System will alert user that employee schedule has not yet been created</li> <li>Use Case terminates</li> </ol>		
<b>Exceptions:</b>	See alternate flows (above)		

<b>Includes:</b>	System
<b>Frequency of Use:</b>	On demand, 1-20 times per week
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions.
<b>Notes and Issues:</b>	N/A

### 1.2.50 Retrieve Browse Employee Schedules

<b>Use Case ID:</b>	1.2.50		
<b>Use Case Name:</b>	Retrieve browse employee schedules		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Supervisor		
<b>Description:</b>	Retrieve list of employee schedules		
<b>Trigger:</b>	User requests to browse list of employee schedules		
<b>Preconditions:</b>	1. Assumptions are true 2. Employee schedule must have already been created		
<b>Postconditions:</b>	1. Supervisors needs of browsing through schedules is satisfied. 2. No information is changed.		
<b>Normal Flow:</b>	1. User chooses the option to browse employee schedules 2. System returns the list of employees' schedules 3. User selects a specific employees schedule to view details of 4. User exits program.		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the User is not logged in to the system <ol style="list-style-type: none"> <li>1. System prompts the User to sign in</li> <li>2. Use Case resumes on step 1</li> </ol> 1b. In step 1 of the normal flow, if the system is not available <ol style="list-style-type: none"> <li>1. System will alert user that it is not available and to try again later</li> <li>2. Use Case terminates</li> </ol> 2a. In step 2 of the normal flow, if the employee schedule does not exist <ol style="list-style-type: none"> <li>1. System will alert user that employee schedule has not yet been created</li> <li>2. Use Case terminates</li> </ol>		
<b>Exceptions:</b>	2b. Database issues <ol style="list-style-type: none"> <li>1. The user is informed of the database issue</li> <li>2. The user is returned to the starting screen</li> </ol>		

<b>Includes:</b>	Create, Update, Delete Employee Schedule
<b>Frequency of Use:</b>	On demand, 1-10 times per week
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions. There are active employees with existing schedules.
<b>Notes and Issues:</b>	N/A

### 1.3.23 Update Employee Roles

<b>Use Case ID:</b>	1.3.23		
<b>Use Case Name:</b>	Update employee roles		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-19	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Manager		
<b>Description:</b>	Update/edit employee role(s)		
<b>Trigger:</b>	Manager requests edit employee roles		
<b>Preconditions:</b>	1. Manager must be logged in 2. Employee profile must have already been created		
<b>Postconditions:</b>	1. If logged in and employee profile was already created, and request is submitted successfully, changes to employee role will be made to the system		
<b>Normal Flow:</b>	1. Manager chooses edit employee role 2. Manager chooses employee to edit the role for 3. Manager fills in the change(s) to the role(s) 4. Manager completes the request 5. Employee role is updated in the system		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Already Logged In]</b>	1a. In step 1 of the normal flow, if the Manager is not logged in to the system <ol style="list-style-type: none"> <li>1. System prompts the Manager to sign in</li> <li>2. Use Case resumes on step 1</li> </ol> b. In step 1 of the normal flow, if the system is not available <ol style="list-style-type: none"> <li>1. System will alert user that it is not available and to try again later</li> <li>2. Use Case terminates</li> </ol>		

	<p>2a. In step 2 of the normal flow, if the employee is not active</p> <ol style="list-style-type: none"> <li>1. System will alert user that employee has not yet been created</li> <li>2. Use Case terminates</li> </ol>
<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	Create, Retrieve, Delete/Deactivate Employee Roles
<b>Frequency of Use:</b>	On demand, 1-20 times per week
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions.
<b>Notes and Issues:</b>	N/A

#### 1.3.24 Update Employee Schedule

<b>Use Case ID:</b>	1.3.24		
<b>Use Case Name:</b>	Update employee schedule		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Primary: Supervisor		
<b>Description:</b>	Update/edit employee schedule		
<b>Trigger:</b>	Supervisor/user requests edit employee schedule		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User must be logged in</li> <li>2. Employee schedule must have already been created</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. If logged in and employee schedule was already created, and request is submitted successfully, changes to employee schedule will be made to the system</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User chooses edit employee schedule</li> <li>2. User chooses employee to edit the schedule for</li> <li>3. User fills in the changes to the schedule</li> <li>4. User completes the request</li> <li>5. Employee schedule is updated in the system</li> </ol>		
<b>Alternative Flows:</b>	1a. In step 1 of the normal flow, if the User is not logged in to the system		
<b>[Alternative Flow 1 –</b>	<ol style="list-style-type: none"> <li>1. System prompts the User to sign in</li> <li>2. Use Case resumes on step 1</li> </ol>		

<b>Already Logged In]</b>	
<b>Alternative Flows:</b> <b>[Alternative Flow 2 – System unavailable]</b>	1b. In step 1 of the normal flow, if the system is not available <ul style="list-style-type: none"> <li>1. System will alert user that it is not available and to try again later</li> <li>2. Use Case terminates</li> </ul>
<b>Alternative Flows:</b> <b>[Alternative Flow 3 – Selected employee schedule not yet created]</b>	2a. In step 2 of the normal flow, if the employee is not active <ul style="list-style-type: none"> <li>1. System will alert user that employee schedule has not yet been created</li> <li>2. Use Case terminates</li> </ul>
<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	System
<b>Frequency of Use:</b>	On demand, 1-20 times per week
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	NA

#### 1.4.13 Delete Employee Roles

<b>Use Case ID:</b>	1.4.13		
<b>Use Case Name:</b>	Delete employee roles		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Manager		
<b>Description:</b>	Delete (deactivate) employee roles		
<b>Trigger:</b>	Manager requests to delete employee role records		
<b>Preconditions:</b>	1. Manager must be logged in		
<b>Postconditions:</b>	1. Employee role is deleted		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Manager chooses delete employee roles</li> <li>2. Employee roles are fetched from the system (database)</li> <li>3. Manager chooses to delete a role</li> <li>4. Manager confirms again they would like to delete the role</li> <li>5. Manager submits the change and the employee role is deleted</li> </ol>		

<b>Alternative Flows:</b>	<p>1a. In step 1 of the normal flow, if the Manager is not logged in to the system</p> <ol style="list-style-type: none"> <li>1. System prompts the Manager to sign in</li> <li>2. Use Case resumes on step 1</li> </ol> <p>1b. In step 1 of the normal flow, if the system is not available</p> <ol style="list-style-type: none"> <li>1. System will alert user that it is not available and to try again later</li> <li>2. Use Case terminates</li> </ol> <p>3a. In step 3 of the normal flow, if the employee is not found by the system</p> <ol style="list-style-type: none"> <li>1. System will alert the Manager that no record was found</li> <li>2. Use Case terminates</li> </ol>
<b>Exceptions:</b>	<p>1a. If the manager enters the wrong user name or password:</p> <ol style="list-style-type: none"> <li>1. it will pop up an error message.</li> <li>2. it will ask for the correct user name or password</li> <li>3. it is limited to 3 times trial</li> </ol>
<b>Includes:</b>	System
<b>Frequency of Use:</b>	On demand, 1-5 times per year
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions.
<b>Notes and Issues:</b>	Do we need to send a list of deactivated employees to Manager email?

#### 1.4.15 Delete Schedules

<b>Use Case ID:</b>	1.4.15		
<b>Use Case Name:</b>	Delete schedules		
<b>Created By:</b>	Brandon Beltz	<b>Last Updated By:</b>	Gunardi Saputra
<b>Date Created:</b>	2018-09-19	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Primary: Supervisor		
<b>Description:</b>	Delete (archive) schedules		
<b>Trigger:</b>	User requests to delete (archive) schedule		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Assumptions are true</li> <li>2. There must be schedules to delete</li> </ol>		

<b>Postconditions:</b>	1. If logged in and request is submitted successfully (and schedule exists), schedule is deleted (archived)
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>User chooses delete schedules</li> <li>Employee schedules are fetched from the system</li> <li>User chooses to delete a schedule</li> <li>User confirms again they would like to delete the schedule</li> <li>User submits the change and the schedule is deleted (archived)</li> </ol>
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Not Logged In]</b>	<ol style="list-style-type: none"> <li>In step 1 of the normal flow, if the User is not logged in to the system             <ol style="list-style-type: none"> <li>System prompts the User to sign in</li> <li>Use Case resumes on step 1</li> </ol> </li> <li>In step 1 of the normal flow, if there are no schedules to delete             <ol style="list-style-type: none"> <li>System will alert the user that no schedules are available to delete</li> <li>Use Case terminates</li> </ol> </li> <li>In step 1 of the normal flow, if the system is not available             <ol style="list-style-type: none"> <li>System will alert user that it is not available and to try again later</li> <li>Use Case terminates</li> </ol> </li> </ol>
<b>Exceptions:</b>	See alternate flows (above)
<b>Includes:</b>	System
<b>Frequency of Use:</b>	On demand, 1-50 times per year
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user knows English. User is logged in. User has permissions.
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>Do we need to store every deleted schedule history?</li> <li>Do we need to send deleted list schedule to the user?</li> </ol>

### 2.2.39 View Total Daily Tips

<b>Use Case ID:</b>	2.2.39		
<b>Use Case Name:</b>	View total daily tips		
<b>Created By:</b>	Jesse Tomash	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-15	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Bartender, Food Worker		
<b>Description:</b>	Bartender views total daily tips		
<b>Trigger:</b>	User selects the option to view total daily tips		
<b>Preconditions:</b>	There were tips received today		

	The assumptions are true
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Daily tips are distributed</li> </ul>
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. The user selects the option to total todays tips</li> <li>2. The system totals the daily tips</li> <li>3. The system displays the total tip amount</li> <li>4. The system shows how the tips were distributed</li> <li>5. The system distributes the tips properly to each worker</li> <li>6. The system displays a prompt saying that the tips were distributed</li> </ul>
<b>Alternative Flows:</b>	There are none currently.
<b>Exceptions:</b>	<p>1a. The system cannot find any tips for the date specified</p> <ul style="list-style-type: none"> <li>1. The system alerts the user that no tips were found for today</li> <li>2. The user acknowledges the systems prompt</li> <li>3. The system returns the user to the previous page.</li> </ul>
<b>Includes:</b>	Create, Update, Delete Tip
<b>Frequency of Use:</b>	At least once per work day
<b>Special Requirements:</b>	None currently
<b>Assumptions:</b>	<p>The user is logged in</p> <p>The user has access</p>
<b>Notes and Issues:</b>	None currently

### 3.2.21 Marketing View Budget

<b>Use Case ID:</b>	3.2.21		
<b>Use Case Name:</b>	Marketing-View Budget		
<b>Created By:</b>	Carlos Arzu	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Marketing Manager or assistant		
<b>Description:</b>	The marketing department needs to be checking constantly their budget, so they can organize the activities and make strategic plans.		
<b>Trigger:</b>			
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Marketing Manager or assistant are logged in.</li> </ul>		
<b>Postconditions:</b>	Marketing Manager has viewed budget of a specified topic.		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Marketing Manager or assistant log in.</li> <li>2. Marketing Manager or assistant select Budget (it will only show the</li> </ul>		

	<p>budget of topic related to the marketing department).</p> <ol style="list-style-type: none"> <li>3. Marketing Manager or assistant can add notes but cannot edit the budget.</li> <li>4. Marketing Manager or assistant close the app.</li> </ol>
<b>Alternative Flows:</b>	<p>2a) If no topic is selected</p> <ol style="list-style-type: none"> <li>1. System prompts Marketing Manager to select a topic first,</li> <li>2. Use Case resumes on step 2 of Normal Flow.</li> </ol>
<b>Exceptions:</b>	<p>2a.Database issues</p> <ol style="list-style-type: none"> <li>1. The user is informed of the database issue</li> <li>2. The user is returned to the starting screen</li> </ol>
<b>Includes</b>	N/A
<b>Frequency of Use:</b>	Used when needed.
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions
<b>Notes and Issues:</b>	N/A

### 3.2.48 Browse Account

<b>Use Case ID:</b>	3.2.48		
<b>Use Case Name:</b>	Browse Account		
<b>Created By:</b>	Austin Berquam	<b>Last Updated By:</b>	Austin Berquam
<b>Date Created:</b>	2018-10-26	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Manager		
<b>Description:</b>	A Manager wants to view the account of a customer/employee		
<b>Trigger:</b>	Managers submits view account		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The assumptions are true.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The account information is open for the manager to view</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Manager requests to see the account of the employee or customer</li> <li>2. System returns account information</li> <li>3. Manager views account information</li> </ol>		
<b>Alternative Flows:</b>	1. In step 1 of the normal flow, if the User is not logged in to the system		

	2. System will prompt User to log in 3. User enters correct login information
<b>Exceptions:</b>	
<b>Includes:</b>	Information of the customer or employee account
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	An account must be made to view an account
<b>Assumptions:</b>	Manager is logged in and accounts exist
<b>Notes and Issues:</b>	1. None

### 3.3.29 Update Account

<b>Use Case ID:</b>	3.3.29		
<b>Use Case Name:</b>	Update Account		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Matt LaMarche
<b>Date Created:</b>	2018-09-16	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Primary Actor: Worker, Guest		
<b>Description:</b>	The Primary Actor wants to update their account information		
<b>Trigger:</b>	An Primary Actor requests to update their account		
<b>Preconditions:</b>	1. The Primary Actor has an account		
<b>Postconditions:</b>	1. The Primary Actor will have had their account details updated		
<b>Normal Flow:</b>	1. Primary Actor requests to update their account 2. System verifies what they can update and shows them what they can change 3. Primary Actor makes the changes they want to make 4. Primary Actor chooses to save the changes they made 5. System updates the Primary Actors account with the provided details		
<b>Alternative Flows:</b>	4a. Primary Actor supplies invalid information Normal Flow in Step 4 1. return to Normal Flow Step 3		
<b>Exceptions:</b>	Do not have permission to update account if attempts to update someone else account		

<b>Includes:</b>	3.2.38 Read Account
<b>Frequency of Use:</b>	As needed
<b>Special Requirements:</b>	None
<b>Assumptions:</b>	Primary Actor is logged in
<b>Notes and Issues:</b>	May need to have an alternate flow for the differences between an employee and a Guest. Currently, I do not know of any differences between the two

#### 6.1.08 Create a Message

<b>Use Case ID:</b>	6.1.08		
<b>Use Case Name:</b>	Create a Message		
<b>Created By:</b>	Gunardi Saputra	<b>Last Updated By:</b>	Gunardi Saputra
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-10-31
<b>Actor:</b>	Primary: Staff		
<b>Description:</b>	Create a message so that Staff can send direct message to Guest, Manager or between Staff.		
<b>Trigger:</b>	Staff wants to communicate directly from the system to Staff or Manager.		
<b>Preconditions:</b>	1. Staff already log in to system 2. Staff choose create message button		
<b>Postconditions:</b>	1. Staff click send button to send the message		
<b>Normal Flow:</b>	1. Staff choose create message button 2. Staff choose recipient: Guest, Staff or (and) manager 3. Staff choose the subject of the message 4. Staff create the email content 5. Staff click send button to send the message		
<b>Alternative Flows:</b>	5a. In step 5 of the normal flow, if the Staff has not finish writing message		
<b>[Alternative Flow 1 – Not in Network]</b>	1. System will prompt Staff to save to draft		
<b>Exceptions:</b>	2a. In step 2 of the normal flow, if the Staff skip choosing the recipient 1. System will prompt Staff to choose the recipient 2. Message to Staff to choose the recipient		

	<p>3a. In step 3 of the normal flow, if the Staff skip choosing the subject</p> <ol style="list-style-type: none"> <li>1. System will prompt Staff to choose the subject</li> <li>2. Message to Staff to choose the subject</li> </ol> <p>4a. In step 4 of the normal flow, if the Staff skip create the email content</p> <ol style="list-style-type: none"> <li>1. System will prompt Staff to write the email</li> <li>2. Message to Staff not to send blank email content</li> </ol>
<b>Includes:</b>	Save in sent folder or save in draft for unfinished or unsent message.
<b>Frequency of Use:</b>	50 per week
<b>Special Requirements:</b>	Contacts limited to registered Guest, Manager and Staff. It cannot add outside contact.
<b>Assumptions:</b>	Staff understand English and follow the message instruction.
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. Do we need to send cc: to every message made to the registered email?</li> </ol>

## Event Operations

The following use cases detail the design relating to the event logistics.

### 2.2.50 Search for a Schedule Event

<b>Use Case ID:</b>	2.2.50		
<b>Use Case Name:</b>	Search for a scheduled event.		
<b>Created By:</b>	Craig Barkley	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Maintenance Personnel		
<b>Description:</b>	Maintenance needs to search for resort information		
<b>Trigger:</b>	Maintenance is searching for information need to complete a work task or to reference account information.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Maintenance personnel should have access to resort management system and be logged in under the correct credentials.</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Maintenance searches event details.</li> </ol>		

	2. Maintenance can prepare and maintain any necessary jobs for the event to proceed without issue based on information found in the search.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Maintenance logs in to resort CMS.</li> <li>2. Maintenance enters username or email.</li> <li>3. Maintenance enters password.</li> <li>4. System validates Maintenance</li> <li>5. Personnel accesses the search view.</li> <li>6. Maintenance makes a search query.</li> <li>7. Search returns results</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1a. In step 4 of the normal flow, if the Maintenance is not online.</li> <li>1. Maintenance can request Search Details print off from resort or events manager.</li> <li>2. Use Case resumes on step 5</li> </ol>
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>2a. In step 2 of the normal flow, if the Personnel enters an invalid Password</li> <li>1. Transaction is disapproved</li> <li>2. Message to Maintenance to re-enter Password</li> <li>3. Maintenance enters correct Password</li> <li>4. Use Case resumes on step 4 of normal flow</li> </ol>
<b>Includes:</b>	Maintenance could also email event details. The Manager would also have access to this exact flow since they may be printing out the Event details to those who do not have access to email.
<b>Frequency of Use:</b>	This Use Case will be executed for Each Event Unless Maintenance has Access to online event to retrieve data themselves.
<b>Special Requirements:</b>	Maintenance will need to have access to the internet to access event details.
<b>Assumptions:</b>	That Maintenance will have access to the internet and that the Catering manager will follow procedure and ensure that Maintenance has accessed the search and view details.
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. They can print off any current event details that have been paid for in full.</li> </ol>

### *3.1.03 Create Event*

<b>Use Case ID:</b>	3.1.03		
<b>Use Case Name:</b>	Create Event		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Primary Actor: Event Manager		

<b>Description:</b>	Someone submits an Event Request to the Event Manager through another part of our system. The Event Manager makes sure the Event Request is valid and can accept or reject the request. If he accepts the Event Request a new Event is created and we need to make sure appropriate personnel and departments are contacted
<b>Trigger:</b>	An Event Request was submitted
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. An Event Request was submitted for the Event Managers Approval</li> <li>2. Event Manager is logged in</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. A new Event has been Created.</li> <li>2. Appropriate forms have been generated</li> <li>3. Appropriate</li> </ol>
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Event Manager reviews an Event Request and Approves the request</li> <li>2. If the Location is on site, add the event to the building(s) schedule(s)</li> <li>3. If the Event wants us to cater for them create a form with the menu and quantity information and submit it to Catering</li> <li>4. If the Event is on site create an employee estimate for the number of employees needed to work the event</li> <li>5. Prompt the Event Manager to reach out to the talent to see if we can book them for the event</li> </ol>
<b>Alternative Flows:</b>	<p>1a. In Step 1 of the Normal flow the Event Manager can Reject the Event Request.</p> <ol style="list-style-type: none"> <li>1. System will prompt the Event Manager to provide a reason why the Event Request was rejected</li> </ol> <p>2a. In step 2 of the Normal Flow the Event may not be on site</p> <ol style="list-style-type: none"> <li>1. Set the Event to an offsite location and add the event to a general off-site schedule</li> </ol> <p>4a. If the Event is off site prompt the Event manager for the number of estimated employees required</p> <p>Note: Insert a new row for each distinctive alternative flow. ]</p>
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. The Event request can be customizable based on what the event requires</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	As needed
<b>Special Requirements:</b>	None
<b>Assumptions:</b>	The user is logged in. The user has access
<b>Notes and Issues:</b>	None

### 3.1.04 Create Event Request

<b>Use Case ID:</b>	3.1.04		
<b>Use Case Name:</b>	Create Event		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Event Manager		
<b>Description:</b>	An Event Manager wants to create an Event		
<b>Trigger:</b>	New event needs to be booked		
<b>Preconditions:</b>	1. Logged In with credentials. 2. Manager has information to book an event 3. System Calendar has open schedule to book event on date.		
<b>Postconditions:</b>	1. A new Event was created for a specific date.		
<b>Normal Flow:</b>	1. Event Manager checks logistics of schedule. 2. Event Manager chooses a building/location 3. Event Manager sets the number of people attending 4. Event Manager chooses the catering options 5. Event Manager specifies who can attend 6. Event Manager specifies whether pets are allowed 7. Event Manager specifies whether children are allowed 8. Event Manager specifies whether Talent is going to perform 9. Event Manager makes notes on additional requirements		
<b>Alternative Flows:</b>	1.. In step 3 of the normal flow, if the event manager does not have access to the guest count they will make note of this for the other departments to see.  2. Personnel can request Event information to be emailed or have a print off from resort or another manager to verify new information.		
<b>Exceptions:</b>	2a. In step 1 of the normal flow, if the event manager adds an event to a previously scheduled time the system will:  1. Creation is disapproved 2. Message to personnel to re-enter date. 3. Personnel enters correct date 4. Use Case resumes on step 2 of normal flow]  1a. Database Error  1. Display error message 2. Return to step 1.		
<b>Includes:</b>	1. Personnel could also email updated event information. The Manager		

	would also have access to this exact flow since they may be printing out employee information details to those who do not have access to email.
<b>Frequency of Use:</b>	This Use Case will be executed for employees Unless personnel have Access to online view and request schedules themselves.
<b>Special Requirements:</b>	Personnel will need to have access to the internet to access event details.
<b>Assumptions:</b>	That Personnel will have access to the internet and that the event manager will follow procedure and ensure that employees have accessed the schedule details.
<b>Notes and Issues:</b>	1. They have corrected permission to edit and save events details.

### *3.1.06 Create Advertising Event*

<b>Use Case ID:</b>	3.1.06		
<b>Use Case Name:</b>	Create Advertising Event		
<b>Created By:</b>	Carlos Arzu	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Primary: Marketing Manager		
<b>Description:</b>	A marketing manager or assistant can create a new event to promote the resort.		
<b>Trigger:</b>	New event with opportunity to advertise the resort.		
<b>Preconditions:</b>	The Marketing Manager or assistant verify the necessary requirements are met to schedule(budget, date, suppliers, other departments and location) of the Event.		
<b>Postconditions:</b>	Manager or assistant schedules event.		
<b>Normal Flow:</b>	1. Marketing Manager or assistant is suitable for advertising the resort. 2. Marketing Manager or assistant schedule the event 3. Marketing Manager lists the materials needed 4. Marketing Manager or assistant list the suppliers who will provide materials.		
<b>Alternative Flows:</b>	3a) If one or more of materials needed are not in stock <ol style="list-style-type: none"> <li>1. System alerts Marketing Manager of materials not being in stock</li> <li>2. Use Case resumes on step 2 of Normal Flow</li> </ol>		
<b>Exceptions:</b>	2a) Database Error <ol style="list-style-type: none"> <li>1. Display database error.</li> </ol>		

	2. Return to step 2
<b>Includes:</b>	1. Marketing-SeeScheduledEvents
<b>Frequency of Use:</b>	Used weekly.
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The event is a good opportunity to promote the resort and attract more visitors. User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

### 3.1.08 Create New Sponsor

<b>Use Case ID:</b>	3.1.08		
<b>Use Case Name:</b>	Create New Sponsor		
<b>Created By:</b>	Carlos Arzu	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-10-30
<b>Actors:</b>	Marketing Manager or assistant		
<b>Description:</b>	A marketing manager or assistant need to have sponsors for events and need to reserve a spot.		
<b>Trigger:</b>	Marketing Manager or assistant when creating or editing an event needs to add sponsor.		
<b>Preconditions:</b>	1. The assumptions are true. 2. Sponsor must have a profile to reserve a spot.		
<b>Postconditions:</b>	New Sponsor will be created.		
<b>Normal Flow:</b>	1. Marketing Manager or assistant login. 2. System validates login credentials. 3. Marketing Manager selects Add Sponsor to Event. 4. System prompts Marketing Manager for name of Sponsor. 5. Marketing Manager inputs name. 6. Marketing Manager inputs details regarding Sponsor. 7. System prompts Marketing Manager to save Sponsor record. 8. Marketing Manager confirms choice. 9. System alerts Marketing Manager of changes saved.		
<b>Alternative Flows:</b>	8a) If Marketing Manager declines to confirm choice by selecting Cancel 1. System confirms choice and does not save Sponsor record 2. Use Case resumes on step 3 of Normal Flow		
<b>Exceptions:</b>	2a) If Marketing Manager enters invalid login credentials		

	<ol style="list-style-type: none"> <li>1. System alerts Marketing Manager of invalid login and prompts them to re-enter login credentials</li> <li>2. Use Case resumes on step 1 of Normal Flow.</li> </ol>
<b>Includes:</b>	Create Event, edit event.
<b>Frequency of Use:</b>	Used as needed
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	Marketing manager or assistant assumes he will have several sponsors and reserves spots. User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

### 3.1.10 Create Account

<b>Use Case ID:</b>	3.1.10		
<b>Use Case Name:</b>	Create Account		
<b>Created By:</b>	Jacob Miller	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-08
<b>Actors:</b>	Performer (Talent)		
<b>Description:</b>	A Talent creates an account		
<b>Trigger:</b>	A Talent requests a new account		
<b>Preconditions:</b>	1. The Talent doesn't have an account		
<b>Postconditions:</b>	1. The Talent has an account		
<b>Normal Flow:</b>	1. The talent selects the option to create an account 2. The system displays the create account form 3. User enters all the necessary information on the form 4. User selects the option to finalize and create account. 5. System validates the form was filled out correctly 6. The system creates the users account and saves it in the database with all other user accounts.		
<b>Alternative Flows:</b>	4a. In Step 3 of the normal flow if the Talent has submitted blank fields <ol style="list-style-type: none"> <li>1. The system alerts the Talent that one or more fields was blank and returns them to the previous screen.</li> </ol>		
<b>Exceptions:</b>	1b. In step 1 of the normal flow, if the User is not a talent <ol style="list-style-type: none"> <li>1. System will alert that login credentials were incorrect and prompt User to re-enter login information</li> <li>2. Step repeats until valid credentials are entered</li> <li>3. Once entered, Use Case resumes on step 2 of normal flow</li> </ol>		
<b>Includes:</b>	Retrieve, Update, Delete Account		

<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	The user is registered as a talent
<b>Assumptions:</b>	The user is a Talent.
<b>Notes and Issues:</b>	Create Account should be universal throughout the system? Also, how can the system verify the user is a talent, if they don't have an existing account?

### 3.2.23 Read Advertising Event

<b>Use Case ID:</b>	3.2.23		
<b>Use Case Name:</b>	Read Advertising Event		
<b>Created By:</b>	Carlos Arzu	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Marketing Manager or assistant		
<b>Description:</b>	The marketing department needs to view the promotional items that have been used and are in inventory.		
<b>Trigger:</b>	Marketing Manager needs to view promotional items.  Marketing Manager selects view promotional items.		
<b>Preconditions:</b>	1. Marketing manager or assistant are logged in.		
<b>Postconditions:</b>	1. Marketing Manager has viewed a list of current promotional items in inventory.		
<b>Normal Flow:</b>	1. Marketing Manager or assistant select Promotional Items. 2. Marketing Manager or assistant can view and request inventory. 3. Marketing Manager or assistant check save if changes were made. 4. Marketing Manager or assistant close app.		
<b>Alternative Flows:</b>	2a) If selected item is currently out of stock  1. System alerts Marketing Manager of items stock and suggests ordering more. 2. Use Case resumes on step 2 of Normal Flow.		
<b>Exceptions:</b>	1a) Database Error  1. Display error message 2. Return to step 1.		
<b>Includes:</b>	Market Events, Create events and edit events		
<b>Frequency of Use:</b>	Use as requested.		

<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions.
<b>Notes and Issues:</b>	Normal flow doesn't make sense?

### 3.2.31 Read Event Request

<b>Use Case ID:</b>	3.2.31		
<b>Use Case Name:</b>	Read Event Request		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Matt LaMarche
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Event Host or Event Manager		
<b>Description:</b>	A Primary Actor wants to view an Event Request		
<b>Trigger:</b>	Primary Actor requests to view an Event Request		
<b>Preconditions:</b>	1. An Event Host has created an Event Request 2. The Event Request has not been approved by an Event Manager yet		
<b>Postconditions:</b>	1. The details of the Event Request will be available to view		
<b>Normal Flow:</b>	1. Primary Actor requests to see the list of active Event Requests 2. System returns all the Event Requests which have not yet been accepted by an Event Manager 3. Primary Actor selects an Event Request from the list 4. System returns the details of the Event Request		
<b>Alternative Flows:</b>	1. In Step 2 of the Normal flow, if the Primary Actor is an Event Host <ol style="list-style-type: none"> <li>System will only return Event requests which the Event Host has created themselves</li> </ol>		
<b>Exceptions:</b>	1. Event Host has no Event Requests available 2. Event Request has been rejected by an Event Manager and is no longer active		
<b>Includes:</b>	3.2.44 Browse Event Requests		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	The Primary Actor is logged into our system.		
<b>Notes and Issues:</b>	None		

### 3.2.35 Read Event

<b>Use Case ID:</b>	3.2.35		
<b>Use Case Name:</b>	Read Event		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Gunardi Saputra
<b>Date Created:</b>	2018-09-25	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Customer		
<b>Description:</b>	An Customer wants to check the details of an Event		
<b>Trigger:</b>	The Customer requests the details for a specific Event		
<b>Preconditions:</b>	1. The Customer can view the details for the Event		
<b>Postconditions:</b>	The Customer will have all the details for the Event which they can see displayed for them		
<b>Normal Flow:</b>	1. Customer needs to create report in the system 2. Customer logs in to resort CMS. 3. Customer username or email. 4. Customer enters password. 5. System validates Customer. 6. Customer selects an Event 7. Customer requests the Event Details for the selected Event 8. System returns Details for the Event which the Customer has permission to view		
<b>Alternative Flows:</b>	1a. In step 4 of the normal flow, if the Customer are not online.  1. Customer can't request Read Event off from resort or have events Customer complete the task in the system.  2. Use Case resumes on step 5		
<b>Exceptions:</b>	2a. In step 2a of the normal flow the link for the details doesn't work  1. The system returns a message that link doesn't work 2. The user acknowledges the message 3. The user is taken to the previous page		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	50 per day		
<b>Special Requirements:</b>	Customer will need to have access to the internet to access view event		
<b>Assumptions:</b>	Needs to be Logged In		
<b>Notes and Issues:</b>	They have corrected permission view details.		

### 3.2.36 Read Setup List

<b>Use Case ID:</b>	3.2.36
---------------------	--------

<b>Use Case Name:</b>	Read Setup List		
<b>Created By:</b>	Caitlin Abelson	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-11	<b>Last Revision Date:</b>	2018-11-08
<b>Actors:</b>	Event Staff		
<b>Description:</b>	Event Staff looks over list of what to set up for an Event		
<b>Trigger:</b>	User selects the option to view setup list		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Event Staff is assigned to the role of Setup Crew</li> <li>2. Event Staff has received the list of what to set up</li> <li>3. The assumptions are true</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Event Staff has the knowledge of what needs setup for the event</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Event Staff logs into account</li> <li>2. Event Staff clicks on Events tab</li> <li>3. Event Staff clicks on Inventory tab</li> <li>4. Event Staff can see details on what inventory was reserved</li> <li>5. Event Staff logs out</li> </ol>		
<b>Alternative Flows:</b>	<p>3a. In step 3 of the normal flow, if Event Staff does not have authorized role to view inventory</p> <ol style="list-style-type: none"> <li>1. System prompts Event Staff that they are not authorized to view inventory</li> <li>2. Normal flow continues at step 5</li> </ol> <p>4a. In step 4 of the normal flow, if no inventory was reserved</p> <ol style="list-style-type: none"> <li>3. Event Staff clicks on reserve inventory</li> <li>4. Inventory form displays</li> <li>5. Event Staff fills out inventory form</li> <li>6. Submits form to Inventory</li> </ol>		
<b>Exceptions:</b>	<p>4a. In step 4 of the alternative flow, if Event Staff submits form with invalid information</p> <ol style="list-style-type: none"> <li>1. System error prompts Event Staff of invalid information</li> <li>2. Event Staff fills out inventory form with valid information</li> <li>3. Event Staff submits form</li> <li>4. Normal flow continues at step 5</li> </ol>		
<b>Includes:</b>	Create, Update, Delete Setup List		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	None currently		
<b>Assumptions:</b>	<p>The user is logged in</p> <p>The user has access</p> <p>An event needing setup exists</p>		

<b>Notes and Issues:</b>	None currently
--------------------------	----------------

### [3.2.37 Read Performance Details](#)

<b>Use Case ID:</b>	3.2.37		
<b>Use Case Name:</b>	Read Performance Details		
<b>Created By:</b>	Caitlin Abelson	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-09-24	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Talent		
<b>Description:</b>	A Talent reads their performance details		
<b>Trigger:</b>	A Talent looks at their performances		
<b>Preconditions:</b>	The Talent has one or more performances they need to view the details of		
<b>Postconditions:</b>	The Talent has read through their performance details		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The Talent clicks on Talent tab</li> <li>2. System shows list of performances that the Talent must choose from</li> <li>3. Talent chooses one of their performances to view</li> <li>4. System displays the details of the performance</li> <li>5. Talent reads the details of their performance</li> </ol>		
<b>Alternative Flows:</b>	<p>2a. If the Talent has no performances</p> <ol style="list-style-type: none"> <li>1. The System alerts the Talent that there are no current performances to view.</li> <li>2. Use Case resumes on step 1 of Normal Flow</li> </ol> <p>4a. If the selected performance has no details</p> <ol style="list-style-type: none"> <li>1. The System alerts the Talent that the specified performance currently does not have any details.</li> <li>1. Use Case resumes on step 3 of Normal Flow</li> </ol>		
<b>Exceptions:</b>	<p>2a. If Database Query fails</p> <ol style="list-style-type: none"> <li>1. System alerts Talent that there has been an error retrieving information. Use Case resumes on step 1 of Normal Flow.</li> </ol>		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	<p>The User is a Talent.</p> <p>The User is logged in</p> <p>The User Has permissions</p>		

<b>Notes and Issues:</b>	N/A
--------------------------	-----

### *3.2.38 Read Account Details*

<b>Use Case ID:</b>	3.2.38		
<b>Use Case Name:</b>	Read Account Details		
<b>Created By:</b>	Caitlin Abelson	<b>Last Updated By:</b>	Gunardi Saputra
<b>Date Created:</b>	2018-09-24	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Talent		
<b>Description:</b>	As a Talent I would like to read the account detail so that I can see all my account detail.		
<b>Trigger:</b>	A Talent reads their account		
<b>Preconditions:</b>	1. The Talent has an account		
<b>Postconditions:</b>	1. The Talent has read through their account		
<b>Normal Flow:</b>	1. The Talent clicks on their account tab 2. The system displays their account information 3. The Talent reads through their account details		
<b>Alternative Flows:</b>	1a. If the user already clicked on the account detail but it takes too long or cannot show the account:  1. It will pop up an error message: "Please try again later!"  1b. If for the second time, showing an error message: "Provide administrator contact for more information."		
<b>Exceptions:</b>	1a. If the user already clicked on the account detail but it takes too long or cannot show the account: it will pop up an error message: "Please try again later!"  1b. If for the second time, showing an error message: "Provide administrator contact for more information."		
<b>Includes:</b>	Retrieving account data from the database.		
<b>Frequency of Use:</b>	100 per week.		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	1. The user is a Talent. 2. The user knows English. 3. The user is logged in 4. The user has permissions		
<b>Notes and Issues:</b>	1. Do we need to ask (show message box) the user for incomplete account detail?  2. Do we need to ask the user to update the account detail before closing the tab account?		

### 3.2.44 Browse Event Request

<b>Use Case ID:</b>	3.2.44		
<b>Use Case Name:</b>	Browse Event Request		
<b>Created By:</b>	Dalton Cleveland	<b>Last Updated By:</b>	Alisa Roehr
<b>Date Created:</b>	2018-10-26	<b>Last Revision Date:</b>	2018-11-04
<b>Actors:</b>	Event Manager		
<b>Description:</b>	An Event Manager wants to look through the list of all available event requests to see which ones they would like to approve		
<b>Trigger:</b>	Event Manager selects to view event requests		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Assumptions are true</li> <li>2. There are available events requests to view</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The event requested is either approved or denied</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Event Manager requests to see the list of all active event requests</li> <li>2. System returns a list of all the current event requests</li> <li>3. Event manager selects a specific event request</li> <li>4. System returns the details of that specific event</li> <li>5. Event Manager decides whether to approve the event, and clicks corresponding choice</li> <li>6. System removes event from request list</li> </ol>		
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. There are no events available to view</li> <li>2. System returns a statement saying there are no current event requests</li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. No available event requests to display</li> </ol>		
<b>Includes:</b>	Create, Update, Delete Request		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	User is logged in. User has permissions.		
<b>Notes and Issues:</b>	None		

### 3.2.45 Read All Events

<b>Use Case ID:</b>	3.2.45		
<b>Use Case Name:</b>	Read All Events		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Wes Richardson

<b>Date Created:</b>	2018-10-02	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary Actor: Guest, Worker, Event Manager		
<b>Description:</b>	The Primary Actor will be able to see a list of all the Events they have permission to see		
<b>Trigger:</b>	A Primary Actor requests to see a list of all the Events (potentially with a filter applied)		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Primary Actor must have permission to see at least one Event</li> <li>2. There must have been at least one Event scheduled</li> </ol>		
<b>Postconditions:</b>	1. Primary Actor can see a list with all the Events they can see		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Primary Actor applies filters(Optional)</li> <li>2. Primary Actor requests to see a list of Events</li> <li>3. System returns all Events which match the permissions and filters</li> </ol>		
<b>Alternative Flows:</b>	3a. No Events found: <ol style="list-style-type: none"> <li>a. Display a message saying no Events found</li> </ol>		
<b>Exceptions:</b>	3a. Database issues <ol style="list-style-type: none"> <li>1. The user is informed of the database issue</li> <li>2. The user is returned to the starting screen</li> </ol>		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	Upon Request		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	At Least one Event Exists The user is logged in. The user has permissions.		
<b>Notes and Issues:</b>	Filters include: By Date, By Location, By Active, By Host, By Catering, or By Performance		

### *3.2.46 Browse Performance*

<b>Use Case ID:</b>	3.2.46		
<b>Use Case Name:</b>	Browse Performance		
<b>Created By:</b>	Jared Greenfield	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-11-04	<b>Last Revision Date:</b>	2018-11-08
<b>Actors:</b>	Primary Actor: Worker Secondary Actor: Customer		
<b>Description:</b>	Allows the actor to view a list of performances.		
<b>Trigger:</b>	The actor clicks the “view performances” option.		

<b>Preconditions:</b>	The Assumptions are true.
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The actor can see the list of performances.</li> <li>2. No information has been changed.</li> </ol>
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The actor clicks the option to "view performances".</li> <li>2. The program displays a list of performances.</li> <li>3. The actor reads through the list of performances</li> <li>4. The user closes the list.</li> </ol>
<b>Alternative Flows:</b>	<p>2a. If there are no performances to view.</p> <ol style="list-style-type: none"> <li>1. A message is displayed that says there are no performances to view.</li> <li>2. The use case resumes on step 1 of the normal flow.</li> </ol>
<b>Exceptions:</b>	<p>1a. If the performance link does not work,</p> <ol style="list-style-type: none"> <li>1. System displays an error message stating that the link isn't working</li> <li>2. System returns the user back to the previous page</li> </ol>
<b>Includes:</b>	Create, Update, Delete Performance
<b>Frequency of Use:</b>	As needed
<b>Special Requirements:</b>	None currently.
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	Does a customer need to be logged in to browse the list of performances?

### 3.2.49 Browse Advertising Event

<b>Use Case ID:</b>	3.2.49		
<b>Use Case Name:</b>	Browse Advertising Event		
<b>Created By:</b>	Austin Berquam	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-10-26	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Manager		
<b>Description:</b>	A Manager wants to view all the current Ad Events		
<b>Trigger:</b>	Managers selects View Advertising Events		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Manager is Logged in</li> </ol>		

<b>Postconditions:</b>	1. The Advertising events are open for the manager to view
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Manager requests to browse the Advertising Events</li> <li>System returns a list of Events</li> <li>Manager selects view details</li> <li>System displays details of specified Event</li> </ol>
<b>Alternative Flows:</b>	<p>2a) If System currently has no events</p> <ol style="list-style-type: none"> <li>System alerts Manager that no events are currently in the System.</li> <li>Use Case resumes on step 1 of Normal Flow</li> </ol>
<b>Exceptions:</b>	N/A
<b>Includes:</b>	Information for all events for Advertising
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	An event must exist to be viewed
<b>Assumptions:</b>	User is logged in. Events exist. The user has permissions.
<b>Notes and Issues:</b>	N/A

### 3.2.52 Browse Setup List

<b>Use Case ID:</b>	3.2.52		
<b>Use Case Name:</b>	Browse Setup List		
<b>Created By:</b>	Caitlin Abelson	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-11	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Event Staff		
<b>Description:</b>	Event Staff looks over list of what to set up for an Event		
<b>Trigger:</b>	Event is coming up on the schedule that Event Staff must setup for.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>Event Staff is assigned to the role of Setup Crew</li> <li>Event Staff has received the list of what to set up</li> <li>The assumptions are true</li> </ol>		
<b>Postconditions:</b>	1. Event Staff sees list		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>Event Staff logs into account</li> <li>Event Staff clicks on Events tab</li> <li>Event Staff clicks on Inventory tab</li> <li>Event Staff can see details on what inventory was reserved</li> <li>Event Staff logs out</li> </ol>		
<b>Alternative Flows:</b>	<p>3a. In step 3 of the normal flow, if Event Staff does not have authorized role to view inventory</p> <ol style="list-style-type: none"> <li>System prompts Event Staff that they are not authorized to view inventory</li> <li>Normal flow continues at step 5</li> </ol>		

	<p>4a. In step 4 of the normal flow, if no inventory was reserved</p> <ol style="list-style-type: none"> <li>3. Event Staff clicks on reserve inventory</li> <li>4. Inventory form displays</li> <li>5. Event Staff fills out inventory form</li> <li>6. Submits form to Inventory</li> </ol>
<b>Exceptions:</b>	<p>4a. In step 4 of the alternative flow, if Event Staff submits form with invalid information</p> <ol style="list-style-type: none"> <li>1. System error prompts Event Staff of invalid information</li> <li>2. Event Staff fills out inventory form with valid information</li> <li>3. Event Staff submits form</li> <li>4. Normal flow continues at step 5</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is logged in. The user has permissions
<b>Notes and Issues:</b>	N/A

### 3.3.15 Update Advertising Event

<b>Use Case ID:</b>	3.3.15		
<b>Use Case Name:</b>	Update Advertising Event		
<b>Created By:</b>	Carlos Arzu	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Marketing Manager or assistant		
<b>Description:</b>	A marketing manager or assistant can edit information about an event.		
<b>Trigger:</b>	Changes, notes or updates for an existing event.		
<b>Preconditions:</b>	<p>Marketing Manager or assistant needs:</p> <ol style="list-style-type: none"> <li>1. Authorization with budget, if applicable for the changes.</li> <li>2. Authorization with Schedule , if applicable for the changes.</li> <li>3. Suppliers confirmation of delivery, if applicable for the changes.</li> <li>4. Update Notes no authorization needed.</li> </ol>		
<b>Postconditions:</b>	<p>Manager or assistant edit event with information about:</p> <ol style="list-style-type: none"> <li>1. Final Expenses.</li> <li>2. Participants and information for database.</li> <li>3. Positive and negative comments of event.</li> </ol>		

<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Marketing Manager or assistant Make the changes needed for the event.</li> <li>2. Marketing Manager or assistant specify to whom the system will let know of the changes.</li> <li>3. Save the event with new updates.</li> </ol>
<b>Alternative Flows:</b>	<p>3a) If the event does not meet the requirements</p> <ol style="list-style-type: none"> <li>1. The System alerts the Manager of invalid requirements on Event</li> <li>2. Use Case resumes on step 1 of Normal Flow</li> </ol>
<b>Exceptions:</b>	N/A
<b>Includes:</b>	Marketing-See Scheduled Events
<b>Frequency of Use:</b>	Used weekly.
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	N/A

### 3.3.26 Update Performance

<b>Use Case ID:</b>	3.3.26		
<b>Use Case Name:</b>	Update Performance		
<b>Created By:</b>	Jacob Miller	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Talent		
<b>Description:</b>	Talent edits one or more performances		
<b>Trigger:</b>	A Talent requests to edit a performance		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The Talent is logged in</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The Talent has edited one or more performances</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The Talent requests to edit a performance</li> <li>2. The system returns a list of performances</li> <li>3. The Talent selects the performance they would like to edit</li> <li>4. The system returns the performance properties form</li> <li>5. The Talent edits one or more fields and submits</li> <li>6. The system updates the form</li> </ol>		
<b>Alternative Flows:</b>	<p>1a. In Step 1 of the normal flow if the Talent has no performances</p> <ol style="list-style-type: none"> <li>1. The system alerts the Talent that they have no performances and returns them to the previous screen</li> <li>2. The Talent acknowledge the Message</li> </ol>		
<b>Exceptions:</b>	N/A		

<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The user is a Talent.
<b>Notes and Issues:</b>	N/A

### *3.3.30 Update Event Request*

<b>Use Case ID:</b>	3.3.30		
<b>Use Case Name:</b>	Update Event Request		
<b>Created By:</b>	Carlos Arzu	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-02	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Event Host		
<b>Description:</b>	Event Host can access the event and update information about the event previous, during and after the event.		
<b>Trigger:</b>	The user clicks on the option to update event request		
<b>Preconditions:</b>	The assumptions are true		
<b>Postconditions:</b>	The event request is updated		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user selects a specific event request</li> <li>2. The system returns the details of that request</li> <li>3. The user makes one or more changes to their request</li> <li>4. The user submits their updated request</li> <li>5. The system saves the request and alerts the user that their changes were made.</li> </ol>		
<b>Alternative Flows:</b>	<p>3a. In step 3 of the normal flow, the user doesn't make any changes before they save.</p> <ol style="list-style-type: none"> <li>1. The system alerts the user that no changes were made</li> <li>2. User accepts the prompt from the system</li> <li>3. System returns the user back to the event request details</li> </ol>		
<b>Exceptions:</b>	<p>3a. In step 3 of the normal flow, the user enters invalid information.</p> <ol style="list-style-type: none"> <li>1. The system alerts the user that there are errors with the information they entered</li> <li>2. User fixes their errors and submits</li> <li>3. System updates event requested alerts user that their changes were successfully made.</li> </ol>		

<b>Includes:</b>	Create, Read, Delete Event Request
<b>Frequency of Use:</b>	Used upon request.
<b>Special Requirements:</b>	None currently
<b>Assumptions:</b>	The user is logged in The user has access
<b>Notes and Issues:</b>	None currently.

### *3.4.10 Delete Scheduled Performance*

<b>Use Case ID:</b>	3.4.10		
<b>Use Case Name:</b>	Delete Scheduled Performance		
<b>Created By:</b>	Jacob Miller	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	2018-09-20	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Talent		
<b>Description:</b>	A Talent Cancels one of their events		
<b>Trigger:</b>	A Talent requests to cancel one of their events		
<b>Preconditions:</b>	1. The Talent is logged in		
<b>Postconditions:</b>	1. The Talent has cancelled one of their events		
<b>Normal Flow:</b>	1. The Talent requests to cancel one of their events 2. The system returns the list of events 3. The Talent selects the event they would like to cancel 4. The system deactivates the selected event		
<b>Alternative Flows:</b>	1a. In Step 1 of the normal flow if the Talent has scheduled events 1. The system alerts the Talent that they have no scheduled events and returns them to the previous screen.		
<b>Exceptions:</b>	N/A		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	User is logged in. User has permissions.		
<b>Notes and Issues:</b>	N/A		

### 3.4.12 Delete Event Request

<b>Use Case ID:</b>	3.4.12		
<b>Use Case Name:</b>	Delete Event Request		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Matt LaMarche
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Primary Actor: Event Host, Event Manager		
<b>Description:</b>	An Primary Actor wants to delete an Event Request. If the Primary Actor is an Event Host the Event request will be deactivated. If the Primary Actor is an Event Manager, they can delete a deactivated Event Request		
<b>Trigger:</b>	Primary Actor requests to delete an Event Request		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Event Host has created an Event Request</li> <li>2. The Event Request has not been approved by an Event Manager yet</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The Event Request the Primary Actor has selected will be deleted or deactivated</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Primary Actor requests to see the list of active Event Requests</li> <li>2. System returns all the Event Requests which have not yet been accepted by an Event Manager</li> <li>3. Primary Actor selects an Event Request from the list</li> <li>4. Primary Actor requests to delete the Event Request</li> <li>5. System deletes the Event Request</li> </ol>		
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. In Step 2 of the Normal Flow, if the Primary Actor is an Event Host:             <ol style="list-style-type: none"> <li>a. The System will only return Event Requests which the Event Host has Created</li> </ol> </li> <li>2. In Step 5 of the Normal Flow, if the Primary Actor is an Event Host:             <ol style="list-style-type: none"> <li>a. the Event Request gets Deactivated instead of Deleted</li> </ol> </li> </ol>		
<b>Exceptions:</b>			
<b>Includes:</b>	3.2.44 Browse Event Requests		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	Primary Actor is logged in		
<b>Notes and Issues:</b>	We may want a screen exclusively for Deleting Event Requests from a list of deactivated Event Requests for Event Managers only		

### 3.4.14 Delete Events

<b>Use Case ID:</b>	3.4.14
---------------------	--------

<b>Use Case Name:</b>	Delete Events		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Matt LaMarche
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Primary Actor: Event Manager		
<b>Description:</b>	An Event Manager wishes to cancel a scheduled Event		
<b>Trigger:</b>	An Event Request requests to cancel an Event		
<b>Preconditions:</b>	1. Assumptions are true		
<b>Postconditions:</b>	1. A new Event has been Created. 2. Appropriate forms have been generated 3. Appropriate		
<b>Normal Flow:</b>	1. Primary Actor requests to see an Event 2. System returns the Event 3. Primary Actor selects an Event 4. Primary Actor requests to cancel the Event 5. System prompts the Primary Actor to provide a reason for cancellation 6. Primary Actor provides a reason for cancellation 7. System deactivates the Event 8. The Event Host gets notified that the Event has been cancelled and why 9. System updates schedules for all Buildings which had the location reserved are updated 10. System notifies Event Manager of any work schedules that need to be updated 11. System notifies Talent which were scheduled to perform that the Event has been cancelled and why 12. System notifies Catering the Event has been cancelled and why if the Event had a Catering Request sent out 13. System notifies everyone who RSVP'd the Event has been cancelled and why 14. Event Manager Requests to delete the deactivated event 15. If Events end date is more than a month ago, system deletes the Event		
<b>Alternative Flows:</b>	1. In Step 15 in the Normal Flow, if the Event was supposed to end within a month from now the Event cannot be deleted (Business Rule)		
<b>Exceptions:</b>	None		
<b>Includes:</b>	3.2.35 Read Event		
<b>Frequency of Use:</b>	As needed		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	At least one Event exists which has not occurred yet and the Primary Actor is logged in		

<b>Notes and Issues:</b>	None
--------------------------	------

## Service and Appointment

The following use cases detail the design relating to the resort services including valet, dependent care, and resort shuttles.

### 3.2.30 Read Resort Services

<b>Use Case ID:</b>	3.2.30		
<b>Use Case Name:</b>	Read Resort Services		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-16	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Primary: Guest Secondary: Worker		
<b>Description:</b>	A User views a list of services provided by the Resort		
<b>Trigger:</b>	User clicks the option to view a list of services provided by the Resort		
<b>Preconditions:</b>	User is a guest currently booked at the Resort  The assumptions are true		
<b>Postconditions:</b>	User has viewed a list of services provided by the Resort		
<b>Normal Flow:</b>	1. User selects the option to view services 2. The system returns the list of services being held by the resort 3. The user reads through list of services 4. The user closes the window 5. The system returns the user back to the previous page.		
<b>Alternative Flows:</b>	None currently		
<b>Exceptions:</b>	2a. The system cannot find the list of services <ol style="list-style-type: none"> <li>1. The system alerts the user that no services were found</li> <li>2. The user acknowledges the systems prompt</li> <li>3. The system returns the user back to the previous page.</li> </ol>		
<b>Includes:</b>	Create, Update, Delete Resort Services		

<b>Frequency of Use:</b>	As needed
<b>Special Requirements:</b>	None currently
<b>Assumptions:</b>	The user is logged in The user has access
<b>Notes and Issues:</b>	None currently

### 3.2.24 Read Review

<b>Use Case ID:</b>	3.2.24		
<b>Use Case Name:</b>	Read Review		
<b>Created By:</b>	Matt LaMarche	<b>Last Updated By:</b>	Matt LaMarche
<b>Date Created:</b>	2018-09-25	<b>Last Revision Date:</b>	2018-11-06
<b>Actors:</b>	Primary Actor: Guest, Worker, Public User		
<b>Description:</b>	A Primary Actor wants to read a Review for an Event, Performance, or for the Resort		
<b>Trigger:</b>	The Primary Actor requests the details for a specific Review		
<b>Preconditions:</b>	1. At least one Review exists for the list		
<b>Postconditions:</b>	The Primary Actor will be able to read the details of a Review		
<b>Normal Flow:</b>	1. Primary Actor selects an Event 2. Primary Actor requests to view the reviews for the Event 3. System returns a list of all Reviews for the Event 4. Primary Actor selects a Review to view the details of 5. System returns Details for the Review		
<b>Alternative Flows:</b>	1. In Step 1 of the normal flow the Primary Actor selects a Performance instead of an Event <ol style="list-style-type: none"> <li>Primary Actor requests to view the Reviews for the Performance</li> <li>System returns a list of all Reviews for the Performance</li> <li>Return to Step 4 of the Normal Flow</li> </ol> 2. The Primary Actor selects the Resort instead of an Event <ol style="list-style-type: none"> <li>Primary Actor requests to view the Reviews for the Resort</li> <li>System returns a list of all the Reviews for the Resort</li> <li>Return to step 4 of the Normal Flow</li> </ol>		
<b>Exceptions:</b>	3a. If no Reviews exist for the Event, Performance, or Resort <ol style="list-style-type: none"> <li>System</li> </ol>		
<b>Includes:</b>	3.2.53 Browse Reviews 3.2.35 Read Event 3.2.37 Read Performance		

<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The Event, Performance, or Resort exist. The Primary Actor is logged in The user has permissions
<b>Notes and Issues:</b>	We currently do not have a good way of establishing a restaurants information. We will need to accommodate Reviews for that as well.

### 3.2.47 Browse Resort Service

<b>Use Case ID:</b>	3.2.47		
<b>Use Case Name:</b>	Browse Resort Service		
<b>Created By:</b>	Jared Greenfield	<b>Last Updated By:</b>	Gunardi Saputra
<b>Date Created:</b>	2018-11-04	<b>Last Revision Date:</b>	2018-11-04
<b>Actors:</b>	Primary Actor: Customer, Worker		
<b>Description:</b>	Allows the Actor to view services the resort offers.		
<b>Trigger:</b>	The Actor selects the “view services” button.		
<b>Preconditions:</b>	1. The Assumptions are true.		
<b>Postconditions:</b>	1. No data has been changed.		
<b>Normal Flow:</b>	1. The Actor selects the “view services” button.  2. The program displays a list of services that the resort offers for the actor to browse.		
<b>Alternative Flows:</b>	2a. If in step 2 there are no records to display.  b. A message is displayed saying that there are no records to view.		
<b>Exceptions:</b>	1a. If “view service” button does not load well or too long, it will pop up an error message.  2a. If in step 2 there are no records to display.  b. A message is displayed saying that there are no records to view.		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	50 - 100 per day		
<b>Special Requirements:</b>	Do we need to ask customer contact information such as phone number or email address to get news of available service?		
<b>Assumptions:</b>	1. User is logged in. User has permissions.  2. User knows English.		

<b>Notes and Issues:</b>	Is it necessary for user to login to browse resort service?
--------------------------	---

### [3.2.53 Browse Reviews](#)

<b>Use Case ID:</b>	3.2.53		
<b>Use Case Name:</b>	Browse Reviews		
<b>Created By:</b>	Dalton Cleveland	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-10-26	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Customer, Worker, Manager		
<b>Description:</b>	A customer, worker, or manager, wants to browse the list of motel reviews		
<b>Trigger:</b>	The actor selects to browse reviews		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. There are reviews available to view</li> <li>2. The assumptions are true</li> </ol>		
<b>Postconditions:</b>	1. The actors review needs are satisfied		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The actor selects to view list of reviews</li> <li>2. The system returns the list of reviews</li> <li>3. The actor reads through the list of reviews</li> <li>4. The actor closes the window</li> </ol>		
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. There are no reviews to browse through</li> <li>2. The systems return a statement stating that there are no reviews</li> </ol>		
<b>Exceptions:</b>	1. No available reviews		
<b>Includes:</b>	Create, Update, Delete Review		
<b>Frequency of Use:</b>	On demand		
<b>Special Requirements:</b>	None		
<b>Assumptions:</b>	User is logged in. User has permissions.		
<b>Notes and Issues:</b>	None		

### [5.2.27 Retrieve Vehicle by License Plate](#)

<b>Use Case ID:</b>	5.2.27		
<b>Use Case Name:</b>	Retrieve Vehicle by License Plate		
<b>Created By:</b>	James Heim	<b>Last Updated By:</b>	Ben Hanna

<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary Actor: Valet		
<b>Description:</b>	To retrieve the make, model, owner, and location of a parked vehicle.		
<b>Trigger:</b>	Normal Flow Step 1		
<b>Preconditions:</b>	1. Valet has logged into the software.		
<b>Postconditions:</b>	1. Make, model, and location and registered guest are displayed.		
<b>Normal Flow:</b>	1. Driver enters guest's name. 2. Make, model, mileage, guest's name and location are displayed.		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	2a. In step 2 of the normal flow, if no matching license plate is found 1. System will display license plate not found error. 2. Valet acknowledges error. 3. Use Case resumes on step 1.		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	Valet logged into software.		
<b>Notes and Issues:</b>	N/A		

### *5.2.28 Retrieve Vehicle by Guest Name*

<b>Use Case ID:</b>	5.2.28		
<b>Use Case Name:</b>	Retrieve Vehicle by Guest Name.		
<b>Created By:</b>	James Heim	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary Actor: Valet		
<b>Description:</b>	To retrieve the make, model, owner, and location of a parked vehicle.		
<b>Trigger:</b>	Normal Flow Step 1		

<b>Preconditions:</b>	1. Valet has logged into the software. 2. assumptions are true
<b>Postconditions:</b>	1. Make, model, and location and registered guest are displayed.
<b>Normal Flow:</b>	1. Driver enters guest's name. 2. System displays list of records matching entered name where make, model, mileage, guest's name and location are displayed.
<b>Alternative Flows:</b>	1a) Driver enters wrong name
<b>Exceptions:</b>	2a. In step 2 of the normal flow, if no matching guest name is found 1. System will display license plate not found error. 2. Valet acknowledges error. 3. Use Case resumes on step 1.
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	NA
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	NA

#### 5.2.37 Retrieve Shuttle Vehicle by Vin

<b>Use Case ID:</b>	5.2.37		
<b>Use Case Name:</b>	Retrieve Shuttle Vehicle by VIN		
<b>Created By:</b>	James Heim	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Shuttle Driver		
<b>Description:</b>	To retrieve the VIN, license plate, make, model, mileage, and location of a shuttle vehicle.		
<b>Trigger:</b>	The user selects the option to retrieve shuttle by VIN		
<b>Preconditions:</b>	1. Shuttle Driver has logged into the software.		
<b>Postconditions:</b>	1. Shuttle's details are displayed.		
<b>Normal Flow:</b>	1. User selects the option to retrieve shuttle by VIN 2. User enters the vehicle's VIN.		

	<ol style="list-style-type: none"> <li>3. System searches through the list of vehicles records and returns the shuttle with matching VIN</li> <li>4. User selects the desired shuttle</li> <li>5. System displays the details of the selected shuttle</li> <li>6. User views shuttle details</li> <li>7. User closes the shuttle details window</li> <li>8. System returns the user to the previous page</li> </ol>
<b>Alternative Flows:</b>	There are no alternative flows currently
<b>Exceptions:</b>	<p>3a. In step 3 of the normal flow, if no matching VIN is found</p> <ol style="list-style-type: none"> <li>1. System alerts the user that no matching shuttle was found</li> <li>2. User acknowledges error.</li> <li>3. Use Case resumes on step 1.</li> </ol> <p>5a. In step 5 of the normal flow, the details for the selected shuttle could not be found</p> <ol style="list-style-type: none"> <li>1. System alerts the user that no shuttle details were found</li> <li>2. User acknowledges error.</li> <li>3. Use Case resumes on step 1.</li> </ol>
<b>Includes:</b>	Create, Update, Delete Shuttle By ID
<b>Frequency of Use:</b>	As needed
<b>Special Requirements:</b>	No special requirements.
<b>Assumptions:</b>	<p>The user is logged in</p> <p>The user has access</p>
<b>Notes and Issues:</b>	No issues currently.

#### 5.2.38 Retrieve Shuttle Vehicle by License Plate

<b>Use Case ID:</b>	5.2.38		
<b>Use Case Name:</b>	Retrieve Shuttle Vehicle by License Plate		
<b>Created By:</b>	James Heim	<b>Last Updated By:</b>	James Heim
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-09-13
<b>Actors:</b>	Shuttle Driver		
<b>Description:</b>	To retrieve the VIN, license plate, make, model, mileage, and location of a		

	shuttle vehicle.
<b>Trigger:</b>	Normal Flow Step 1.
<b>Preconditions:</b>	1. Shuttle Driver has logged into the software.
<b>Postconditions:</b>	1. Shuttle's details are displayed.
<b>Normal Flow:</b>	1. Driver enters license plate number 2. Shuttle details are displayed.
<b>Alternative Flows:</b>	2a. In step 2 of the normal flow, if no matching license plate is found 1. System will display license plate not found error. 2. Driver acknowledges error. 3. Use Case resumes on step 1.
<b>Exceptions:</b>	N/A
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	50 per day.
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	N/A

### 5.3.12 Update Vehicle's Parking Spot

<b>Use Case ID:</b>	5.3.12		
<b>Use Case Name:</b>	Update Vehicle's Parking Spot		
<b>Created By:</b>	James Heim	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary Actor: Valet		
<b>Description:</b>	To update the parking spot of a guest's vehicle.		
<b>Trigger:</b>	Vehicle is moved by guest or valet.		
<b>Preconditions:</b>	1. Valet is logged into software.		

<b>Postconditions:</b>	1. Vehicle location is updated.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Valet searches for and selects vehicle. (inclusion)</li> <li>2. Valet clicks update parking spot.</li> <li>3. System allows Valet to edit text.</li> <li>4. Valet types new location.</li> </ol>
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	N/A
<b>Includes:</b>	<p>5.2.27 Retrieve Vehicle by License Plate</p> <p>5.2.28 Retrieve Vehicle by Driver Name</p>
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	No assumptions.
<b>Notes and Issues:</b>	N/A

### *5.3.16 Update Shuttle Vehicle Status*

<b>Use Case ID:</b>	5.3.16		
<b>Use Case Name:</b>	Update Shuttle Vehicle Status		
<b>Created By:</b>	James Heim	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Shuttle Driver		
<b>Description:</b>	I can mark a vehicle as being in available, in use, in service, or decommissioned.		
<b>Trigger:</b>	Normal Flow Step 1.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Shuttle Driver has logged into the software.</li> <li>2. Shuttle Driver has record already pulled up.</li> </ol>		
<b>Postconditions:</b>	1. Shuttle's status is updated.		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Driver selects update vehicle status.</li> <li>2. System displays dropdown of statuses.</li> <li>3. Driver selects new status and clicks submit.</li> </ol>		
<b>Alternative Flows:</b>			

<b>Exceptions:</b>	1a. Database issues <ul style="list-style-type: none"> <li>1. The system informs the user of the issue</li> <li>2. The user is returned to the Vehicle Details Page</li> </ul>
<b>Includes:</b>	View Retrieve Shuttle Vehicle by VIN  View Retrieve Shuttle Vehicle by License Plate
<b>Frequency of Use:</b>	5 Times a Year
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 5.4.05 Deactivate Shuttle Vehicle

<b>Use Case ID:</b>	5.4.05		
<b>Use Case Name:</b>	Deactivate Shuttle Vehicle		
<b>Created By:</b>	James Heim	<b>Last Updated By:</b>	Wes Richardson
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Shuttle Driver		
<b>Description:</b>	Vehicle is removed from the inventory but kept as an archive.		
<b>Trigger:</b>	User is logged in		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>1. Shuttle Driver has logged into the software.</li> <li>2. Shuttle Driver has record already pulled up.</li> </ul>		
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>1. Shuttle's is marked as deactivated.</li> </ul>		
<b>Normal Flow:</b>	<ul style="list-style-type: none"> <li>1. Driver selects Deactivate Vehicle.</li> <li>2. System requests user confirm they want to delete the vehicle</li> <li>3. Driver enters password and clicks submit.</li> </ul>		
<b>Alternative Flows:</b>	No alternate flow.		
<b>Exceptions:</b>	1a.Database issues <ul style="list-style-type: none"> <li>1. The user is informed what the issue is</li> <li>2. The user is returned to the Vehicle List Page</li> </ul>		
<b>Includes:</b>	View List of Vehicles		

	View Vehicle Details
<b>Frequency of Use:</b>	A dozen times per year.
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	No issues.

#### 6.2.05 View List of Children

<b>Use Case ID:</b>	6.2.05		
<b>Use Case Name:</b>	View list of children		
<b>Created By:</b>	Tiona White	<b>Last Updated By:</b>	Jeff Stobb Wes Richardson
<b>Date Created:</b>	2018-09-15	<b>Last Revision Date:</b>	2018-09-21 2018-11-07
<b>Actors:</b>	Manager		
<b>Description:</b>	Manager can see the list of children in the daycare at any given time and the lists of children registered for each event		
<b>Trigger:</b>	Manager clicks child list		
<b>Preconditions:</b>	1. Manager is already logged in		
<b>Postconditions:</b>	1. Manager can now see the lists of children		
<b>Normal Flow:</b>	1. Manager clicks child list 2. Program displays the list options (all current events, or daycare) 3. Manager chooses daycare or whatever event they want 4. Program displays the list of children requested		
<b>Alternative Flows:</b> <b>[Alternative Flow 1 – Not in Network]</b>	None		
<b>Exceptions:</b>	1a. Database issues 1. The database cannot retrieve the information 2. The user is informed of the issue and returned to the starting screen		

<b>Includes:</b>	None
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	Child list button with options of the different lists of children is needed
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

#### 6.2.06 View Child Details

<b>Use Case ID:</b>	6.2.06		
<b>Use Case Name:</b>	View child details		
<b>Created By:</b>	Tiona White	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-15	<b>Last Revision Date:</b>	2018-11-15
<b>Actors:</b>	Manager		
<b>Description:</b>	Manager can view a child's details (personal information and parent information)		
<b>Trigger:</b>	Manager clicks on a child's name to expand their details		
<b>Preconditions:</b>	1. Manager is already viewing a list of children 2. The assumptions are true		
<b>Postconditions:</b>	1. Manager can view the selected child's details		
<b>Normal Flow:</b>	1. Manager is viewing list of children at the resort 2. Manager clicks the option to view details of a child 3. System returns the details of the selected child 4. Manager views details of the child 5. Manager closes the window 6. System returns the user back to the list of children		
<b>Alternative Flows:</b>	There are none currently.		
<b>Exceptions:</b>	1a. The system cannot find the list of children to view 1. The system alerts the user that no children were found 2. The user acknowledges the systems prompt 3. The system returns the user back to the list of children		
<b>Includes:</b>	Create, Update, Delete Child Details		
<b>Frequency of Use:</b>	As needed		
<b>Special Requirements:</b>	Childs name in list must be expandable to show details		

<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	None currently

#### *6.2.23 Update Dependent Record*

<b>Use Case ID:</b>	6.2.23		
<b>Use Case Name:</b>	Update Dependent Record		
<b>Created By:</b>	Gunardi Saputra	<b>Last Updated By:</b>	Gunardi Saputra
<b>Date Created:</b>	2018-10-26	<b>Last Revision Date:</b>	2018-10-31
<b>Actors:</b>	Primary: Guest		
<b>Description:</b>	Guest can update dependent record on the system		
<b>Trigger:</b>	Guest want to see updated record in the system		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Guest already log in</li> <li>2. Choose account profile</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Updated record dependent information in the system.</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Guest retrieve old record information for the dependent</li> <li>2. Choose to edit</li> <li>3. Update new record information for the dependent</li> <li>4. Confirm the new record : save or cancel</li> </ol>		
<b>Alternative Flows:</b>	<p>1a. In step 1 of the normal flow, if the Guest does not have an old record</p> <ol style="list-style-type: none"> <li>1. System will prompt Guest to type in new record</li> <li>2. Confirm the new record: save or cancel.</li> </ol> <p>3a. In step 3 of the normal flow, if the Staff skip choosing the subject</p> <ol style="list-style-type: none"> <li>1. System will prompt Staff to choose the subject</li> <li>2. Message to Staff to choose the subject</li> </ol> <p>4a. In step 4 of the normal flow, if the Staff skip create the email content</p> <ol style="list-style-type: none"> <li>1. System will prompt Staff to write the email</li> <li>2. Message to Staff not to send blank email content</li> </ol>		

<b>Exceptions:</b>	4a. Guest forget to save the new record 1. Message to user to save or cancel the new entry
<b>Includes:</b>	Save updated record in the system.
<b>Frequency of Use:</b>	~30 times a day
<b>Special Requirements:</b>	Do we need to remind the Guest to update dependent record any certain period?
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	<ul style="list-style-type: none"> <li>Do we need to send notice email that update dependent record success or fail?</li> </ul>

#### *7.1.01 Add a New Dependent*

<b>Use Case ID:</b>	7.1.01		
<b>Use Case Name:</b>	Add a new dependent		
<b>Created By:</b>	Austin Delaney	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-10	<b>Last Revision Date:</b>	2018-10-11
<b>Actors:</b>	Primary: Pet staff Secondary: Pet receptionist, Pet manager, Customer		
<b>Description:</b>	The actor adds a pet to the system.		
<b>Trigger:</b>	Actor would like to add a pet to the system.		
<b>Preconditions:</b>	1. A “pet add form” is active 2. The pet is not already in the system 3. The user is logged in and has proper permissions		
<b>Postconditions:</b>	1. The pet’s information is now in the system		
<b>Normal Flow:</b>	1. User inputs pet information to “add pet form” 2. Submit form 3. Verify all fields of the form to ensure valid entry 4. Create pet record in system 5. Close the form 6. Display verification message of successful entry		
<b>Alternative Flows:</b>	2a. In step two of the normal flow, if the user opts to cancel the action 1. Exit the flow		

	<p>2b. In step two of the normal flow, if the user opts to “clear form”</p> <ol style="list-style-type: none"> <li>1. Normal flow resumes on step 1</li> </ol> <p>3a. An invalid entry has been detected</p> <ol style="list-style-type: none"> <li>1. System prompts for valid input</li> <li>2. User acknowledges prompt</li> <li>3. Normal flow resumes on step 1</li> </ol>
<b>Exceptions:</b>	<p>4a. System is unresponsive and/or record was not created</p> <ol style="list-style-type: none"> <li>1. Message to user explaining error</li> <li>2. Normal flow resumes at the end of step 1, with all data still in the form</li> </ol>
<b>Includes:</b>	NA
<b>Frequency of Use:</b>	~30 times a day
<b>Special Requirements:</b>	NA
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	<ul style="list-style-type: none"> <li>• Could probably be expanded into another use case, separate for customers/employees</li> <li>• Should we expand the flow to include each field?</li> <li>• Former Description: Allows a user to interact with the system and create a profile in the software system to represent a customer’s pet. This creation can be done at the front desk by a staff member, receptionist, or manager, or via a mobile interface by a customer.</li> </ul>

#### 7.1.03 Sign Pets Up for Grooming Appointments

<b>Use Case ID:</b>	7.1.03		
<b>Use Case Name:</b>	Sign pets up for grooming appointments		
<b>Created By:</b>	Austin Delaney	<b>Last Updated By:</b>	Austin Delaney
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-02
<b>Actors:</b>	Pet Manager, Pet Worker, Pet Receptionist, Guest		
<b>Description:</b>	User creates a record of a scheduled appointment with the vet technician		
<b>Trigger:</b>	It is determined that the pet in question is needing some sort of medical attention		

<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Pet must be listed in the system</li> <li>2. User must be logged into the system with the appropriate permissions</li> <li>3. From a list of presented options, the user has selected to “vet tech appointment signup”</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. An appointment record has been created with the pets' information and appointment time</li> </ol>
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The system will prompt to select the pet which the user would like to select</li> <li>2. The system will request the preferred time and date of the appointment</li> <li>3. The system will then best determine which available vet tech is best suited for the pet and meeting time</li> <li>4. System will prompt a decision for selected vet tech</li> <li>5. System will prompt user to confirm basic medical information about pet</li> <li>6. System will prompt user to confirm basic pet information, appointment time, and vet tech name</li> <li>7. Create record</li> <li>8. Store record</li> <li>9. Send user back to previous menu(s)</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>4a. Chosen vet tech is deemed unsuitable by user <ol style="list-style-type: none"> <li>1. Show a list of all techs, prompting which the user would prefer</li> <li>2. New form like step 2, allowing user to choose preferred day of appt then displaying times available to that vet tech</li> <li>3. Resume on step 5 of normal</li> </ol> </li> <li>4b. User chooses to return to date select <ol style="list-style-type: none"> <li>1. Resume normal flow on step 2</li> </ol> </li> <li>4c. User opts to cancel the appointment creation <ol style="list-style-type: none"> <li>1. Exit the flow</li> </ol> </li> <li>5a. Medical information about pet is incorrect <ol style="list-style-type: none"> <li>1. Send to an update medical record event</li> </ol> </li> </ol>
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. If we don't have veterinarian records, we can't do anything.</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	At most probably 10 times/day
<b>Special Requirements:</b>	The pet must be set up in database first.
<b>Assumptions:</b>	<ul style="list-style-type: none"> <li>• Assumed that in step 2, the system will output some sort of notification when the chosen date/time is not available or there are</li> </ul>

	no vet techs open at that time.
<b>Notes and Issues:</b>	1. Step one will probably be from a list with an option for pet ID input, so that people who aren't at the resort yet can book their pet for a checkup when they get there, while people who are at the resort can just boom click an item in the list.

#### *7.2.14 Groomer Appointment Book*

<b>Use Case ID:</b>	7.2.14		
<b>Use Case Name:</b>	Groomer Appointment Book		
<b>Created By:</b>	Cody Herb	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-28
<b>Actors:</b>	Groomer Customer		
<b>Description:</b>	Used to look at the Groomers appointment book		
<b>Trigger:</b>	User wants to review their appointment book		
<b>Preconditions:</b>	1. User must be logged in 2. User must be a groomer		
<b>Postconditions:</b>	System will display the Groomers appointment book		
<b>Normal Flow:</b>	1. User will select appointments 2. System will search for the users' appointment list 3. System will display user appointments listed next appointment to furthest appointment		
<b>Alternative Flows:</b>	N/A		
<b>Exceptions:</b>	2a. If system is unable to locate appointment list:  1. User will get message "Unable to find appointment book. Please try again later." 2. Return to follow step 1		
<b>Includes:</b>	NA		
<b>Frequency of Use:</b>	As needed		
<b>Special Requirements:</b>	NA		
<b>Assumptions:</b>	User is a groomer. User is logged in. User has permissions.		
<b>Notes and Issues:</b>	NA		

### 7.2.16 Show Appointment Details

<b>Use Case ID:</b>	7.2.16		
<b>Use Case Name:</b>	Show Appointment Details		
<b>Created By:</b>	Cody Herb	<b>Last Updated By:</b>	Jeff Stobb Wes Richardson
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-10-26 2018-11-07
<b>Actors:</b>	Groomer		
<b>Description:</b>	Used to look at the details of groomer appointment book		
<b>Trigger:</b>	User selects to view appointment details		
<b>Preconditions:</b>	The user is logged in		
<b>Postconditions:</b>	System will show specified groomer appointment details		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User will select appointments</li> <li>2. System will search for the users' appointment list</li> <li>3. User will select the appointment details</li> </ol>		
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. There are no appointments to view</li> <li>2. System returns statement that there are no appointments to return</li> </ol>		
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. If system is unable to locate appointment details:</li> <li>2. User will get message "Unable to find appointment details. Please try again later"</li> <li>3. Return to follow step 1</li> </ol>		
<b>Includes:</b>	Create, Update, Delete Appointment		
<b>Frequency of Use:</b>	As needed		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	User is logged in. User has access		
<b>Notes and Issues:</b>	N/A		

### 7.2.21 Browse Pets

<b>Use Case ID:</b>	7.2.21		
<b>Name:</b>	Browse Pets		
<b>Created By:</b>	Ben Hanna	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	9-11-2018	<b>Last Revision Date:</b>	2018-11-02

<b>Actors:</b>	Pet Worker
<b>Description:</b>	User wishes to browse through the list of pets
<b>Trigger:</b>	User requests to browse through list of pets
<b>Preconditions:</b>	User is logged in. User has permissions.
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. User is viewing the pets of a specified status</li> </ol>
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User selects option to browse through pets</li> <li>2. The system returns the list of pets to the user</li> <li>3. The user views the list of pets with the specified status</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. There are no available pets to browse through</li> <li>2. System returns a statement stating there are no pets to view</li> </ol>
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. No available pets to view</li> </ol>
<b>Includes:</b>	7.2.03 - View Status of all pets
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	None
<b>Assumptions:</b>	None
<b>Notes and Issues:</b>	None

#### 7.3.14 Change an Appointment Details

<b>Use Case ID:</b>	7.3.14		
<b>Use Case Name:</b>	Change an appointment details		
<b>Created By:</b>	Ben Hanna	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	2018-09-15	<b>Last Revision Date:</b>	2018-11-02

	Pet Receptionist
<b>Actors:</b>	
<b>Description:</b>	An appointment details need to be changed
<b>Trigger:</b>	User selects to change appointment details.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. At least 1 appointment must be in system for the Customer or dependent.</li> <li>2. The assumptions are true.</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Pet's appointment details have been updated</li> </ol>
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User selects the Customer that the appointment is to be edited.</li> <li>2. User selects the appointment they wish to edit the details of.</li> <li>3. User edits the appointment.</li> <li>4. User clicks Confirm.</li> <li>5. The appointment as changed is saved.</li> <li>6. The interface returns to the screen where it was selected to edit an appointment.</li> </ol>
<b>Alternative Flows:</b>	<p>4a. User clicks Cancel.</p> <ol style="list-style-type: none"> <li>1. User is asked to confirm they wish to not save the edited appointment.</li> <li>2. User clicks Yes.</li> <li>3. System goes to the page where they selected to edit appointment details.</li> </ol> <p>4b. User clicks Cancel.</p> <ol style="list-style-type: none"> <li>1. User is asked to confirm they wish to not save the edited appointment.</li> <li>2. User clicks Continue Editing.</li> <li>3. User continues in Normal Flow Step 3.</li> </ol> <p>4b. User clicks Cancel.</p> <ol style="list-style-type: none"> <li>1. User is asked to confirm they wish to not save the edited appointment.</li> <li>2. User clicks Save and Quit.</li> <li>3. User continues in Normal Flow Step 5.</li> </ol>

<b>Exceptions:</b>	4a. Program can't access appointments server. 1. Throw error message 2. Return to step 1.
<b>Includes:</b>	7.2.14 Look at appointment book (personal) (GROOM) 7.2.15 Look at appointment book (personal) (VET) 7.2.16 Details for appointment book (GROOM) 7.2.17 Details for appointment book (VET)  Scheduling system,
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions. Customer has appointment already in the schedule.
<b>Notes and Issues:</b>	N/A

#### 7.4.05 Delete Appointment

<b>Use Case ID:</b>	7.4.05		
<b>Use Case Name:</b>	Delete Appointment		
<b>Created By:</b>	Cody Herb	<b>Last Updated By:</b>	Ben Hanna Wes Richardson
<b>Date Created:</b>	2018-09-11	<b>Last Revision Date:</b>	2018-11-02 2018-11-07
<b>Actors:</b>	Primary Actor(s): Worker		
<b>Description:</b>	User wishes to delete/cancel appointment.		
<b>Trigger:</b>	User selects to delete/cancel appointment.		
<b>Preconditions:</b>	1. User must be logged in		

	2. At least 1 appointment must be in system for the Customer.
<b>Postconditions:</b>	1. Appointment is deleted/cancelled.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User selects the Customer that the appointment is to be cancelled from.</li> <li>2. User selects the appointment they wish to delete.</li> <li>3. System will prompt the User if they are sure they want to delete the record</li> <li>4. User selects Yes</li> <li>5. Record is deleted</li> <li>6. The interface returns to the screen where it was selected to delete/cancel an appointment.</li> </ol>
<b>Alternative Flows:</b>	<p>1a. In step 1 of the normal flow, if the User is the Customer</p> <ol style="list-style-type: none"> <li>1. The System automatically selects that the Customer for the appointment to be deleted is the User.</li> <li>2. Use Case resumes on step 2</li> </ol> <p>4a. In step 3 of the normal flow, if the User selects No</p> <ol style="list-style-type: none"> <li>1. The action is cancelled, and the appointment is not deleted/cancelled.</li> <li>2. Use Case resumes on step 6</li> </ol>
<b>Exceptions:</b>	N/A
<b>Includes:</b>	Scheduling system,
<b>Frequency of Use:</b>	8 per week
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

## Product Operations

The following use cases detail the design relating to the resorts shops and restaurants and the products offered.

### 1.2.39 View Products

<b>Use Case ID:</b>	1.2.39
---------------------	--------

<b>Use Case Name:</b>	View Products		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Matthew Hill
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-07
<b>Actors:</b>	Customer		
<b>Description:</b>	Table Assigner wishes to view product list/display		
<b>Trigger:</b>	User selects product display option from the interface		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The assumptions are true</li> <li>2. User has access to Customer Options</li> </ol>		
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Product list is displayed</li> </ol>		
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User selects View Products</li> <li>2. System displays a list of products</li> <li>3. User selects a product from the list</li> <li>4. System displays details of the selected product</li> </ol>		
<b>Alternative Flows:</b>	4a) If selected product has no details in the system. <ol style="list-style-type: none"> <li>1. System does not display any details when the product is clicked on</li> <li>2. Use Case resumes on step 3 of Normal Flow</li> </ol>		
<b>Exceptions:</b>	2a) If Query to Database fails <ol style="list-style-type: none"> <li>1. System alerts User that Database Query has failed</li> <li>2. Use Case resumes on step 1 of Normal Flow</li> </ol>		
<b>Includes:</b>	N/A		
<b>Frequency of Use:</b>	On Demand		
<b>Special Requirements:</b>	N/A		
<b>Assumptions:</b>	The Program displays text in the area's native language. User is logged in. User has permissions.		
<b>Notes and Issues:</b>	N/A		

#### 1.2.40 View Cart

<b>Use Case ID:</b>	1.2.40		
<b>Use Case Name:</b>	View Cart		
<b>Created By:</b>	Richard Carroll	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-14	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Primary: Customer		

<b>Description:</b>	Customer wishes to view their cart of potential online purchases
<b>Trigger:</b>	User selects cart option from interface
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User is logged in</li> <li>2. User has access to Customer</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Program is displaying cart</li> </ol>
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. User selects cart option from the interface</li> <li>2. Program displays cart</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. Cart is empty (Normal flow step 2)             <ol style="list-style-type: none"> <li>a. Program displays the cart is blank and exits normal flow</li> </ol> </li> </ol>
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>1. N/A</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	On Demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	The Program displays text in the area's native language
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. N/A</li> </ol>

#### 2.1.21 Create Report

Use Case ID:	2.1.21		
Use Case Name:	Create Report		
Created By:	Eduardo Colon	Last Updated By:	Jeff Stobb
Date Created:	2018-09-18	Last Revision Date:	2018-11-01
Actors:	Admin, database(secondary)		
Description:	The Admin wishes to create a product report in the system, so he can handle to the sales department		
Trigger:	<ol style="list-style-type: none"> <li>1. Admin needs to create report in the system</li> </ol>		

Preconditions:	<ol style="list-style-type: none"> <li>1. Admin must have an active account with report role privileges</li> <li>2. Admin must be logged in</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. Admin will be able to create a product report</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. Admin needs to create report in the system</li> <li>2. Admin logs in to resort CMS.</li> <li>3. Admin username or email.</li> <li>4. Admin enters password.</li> <li>5. System validates Admin.</li> <li>6. System displays a Report Tab</li> <li>7. Admin presses the report tab</li> <li>8. System display a list of report</li> <li>9. Admin presses the Create Report button</li> <li>10. System request to enter a name</li> <li>11. Admin enters the name</li> <li>12. System displays a report with all the product details</li> <li>13. Admin closes report</li> </ol>
Alternative Flows:	<p>1a. In step 4 of the normal flow, if the Admin are not online.</p> <ol style="list-style-type: none"> <li>1. Admin can't request print report off from resort or have events Admin complete the task in the system.</li> <li>2. Use Case resumes on step 5</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>3a. In step 3 of the normal flow, if the Admin enters and invalid Password           <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to Admin to re-enter Password</li> <li>3. Admin enters correct Password</li> <li>4. Use Case resumes on step 5 of normal flow]</li> </ol> </li> </ol>
Includes:	NA

Frequency of Use:	1 -5 times a week
Special Requirements:	Admin will need to have access to the internet to access event print report details.
Assumptions:	NA
Notes and Issues:	They have corrected permission to edit and print report details.

### 2.2.22 Search Item

<b>Use Case ID:</b>	2.2.22		
<b>Use Case Name:</b>	Search Item		
<b>Created By:</b>	Eduardo Colon	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Web Visitor ,Database (Secondary)		
<b>Description:</b>	Web Visitor wishes to search an item for a menu in the system, so he can add it to his chosen list		
<b>Trigger:</b>	1. Web Visitor needs to search for an item in the system		
<b>Preconditions:</b>	1. Web Visitor must have an active account with product role privileges 2. Web Visitor must be logged in		
<b>Postconditions:</b>	1. Web Visitor will be able to search a product to be added to a menu		
<b>Normal Flow:</b>	1. Web Visitor needs to create report in the system 2. Web Visitor logs in to resort CMS. 3. Web Visitor username or email. 4. Web Visitor enters password. 5. System validates Web Visitor. 6. Web Visitor needs to search for an item in the system 7. System displays a search field 8. Web Visitor enter the item to be searched 9. System returns a matching item record		
<b>Alternative Flows:</b>	9a. In step 9 of the normal flow, if there is not matching search record.		
<b>[Alternative Flow 1 Not</b>	1. System displays an error message		

<b>In System]</b>	2. Web Visitor acknowledges the error  3. The System returns to step 8 of the normal flow.
<b>Exceptions:</b>	3a. In step 3 of the normal flow, if the Web Visitor enters an invalid Password  1. Transaction is disapproved  2. Message to Web Visitor to re-enter Password  3. Web Visitor enters correct Password  Use Case resumes on step 5 of normal flow]
<b>Includes:</b>	NA
<b>Frequency of Use:</b>	On Demands: 1-50 times per day
<b>Special Requirements:</b>	Web Visitor will need to have access to the internet to access search item.
<b>Assumptions:</b>	NA
<b>Notes and Issues:</b>	There are no open issues

#### 2.4.13 Delete Item

<b>Use Case ID:</b>	2.4.13		
<b>Use Case Name:</b>	Delete Item		
<b>Created By:</b>	Eduardo Colon	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-18	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Web Visitor		
<b>Description:</b>	Web Visitor wishes to delete an item from a menu in the system, so he can decrease his chosen list items		
<b>Trigger:</b>	1. System Displays a menu list tab		
<b>Preconditions:</b>	1. Web Visitor must have an active account with product role privileges 2. Web Visitor must be logged in		
<b>Postconditions:</b>	1. Web Visitor will be able to delete a product from a menu list		
<b>Normal Flow:</b>	1. Web Visitor logs in to resort CMS. 2. Web Visitor username or email. 3. Web Visitor enters password. 4. System validates Web Visitor 5. System Displays a menu list tab 6. Web Visitor press the menu list tab		

	<p>7. System displays the Web Visitor menu list  8. Web Visitor select menu to edit items from  9. Web Visitor press edit menu to delete the item  10. Web Visitor select the item to be deleted  11. Web Visitor press the delete item to remove the product  12. System responses with a success of the deleted product.</p>
<b>Alternative Flows:</b>	<p>8a. In step 8 of the normal flow, if there is not selected order.</p> <ol style="list-style-type: none"> <li>1. System displays an error message</li> <li>2. Web Visitor acknowledges the error</li> <li>3. The System returns to step 6 of the normal flow</li> </ol>
<b>Exceptions:</b>	<p>3a. In step 3 of the normal flow, if the Web Visitor enters an invalid Password</p> <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to Web Visitor to re-enter Password</li> <li>3. Web Visitor enters correct Password</li> <li>4. Use Case resumes on step 5 of normal flow</li> </ol>
<b>Includes:</b>	NA
<b>Frequency of Use:</b>	1 -20 times a week
<b>Special Requirements:</b>	Web Visitor will need to have access to the internet to access event delete food item details.
<b>Assumptions:</b>	Needs to be logged in. User needs to have permissions.
<b>Notes and Issues:</b>	<p>There are no open issues</p> <p>They have corrected permission to edit and delete food items.</p>

### 3.2.50 Browse Package

<b>Use Case ID:</b>	3.2.50		
<b>Use Case Name:</b>	Browse Package		
<b>Created By:</b>	Matthew Hill	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-10-26	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	Customer		
<b>Description:</b>	View a list of all our packages.		
<b>Trigger:</b>			
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Customer is logged in to the system.</li> <li>2. Customer has a list of packages they need to view.</li> </ol>		

<b>Postconditions:</b>	1. Customer has viewed package(s).
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Customer logs into system.</li> <li>2. System validates login credentials.</li> <li>3. Customer selects Packages tab.</li> <li>4. Customer selects Browse Packages.</li> <li>5. System displays a list of packages.</li> </ol>
<b>Alternative Flows:</b>	<p>2a. In step 2 of the normal flow, if the customer is not in the system, or input invalid login credentials</p> <ol style="list-style-type: none"> <li>1. System prompts user to re-enter login credentials.</li> <li>2. Customer inputs login credentials.</li> <li>3. Use Case resumes on step 2.</li> </ol> <p>5a. In step 5 of the normal flow, if there are currently no packages to view in the system</p> <ol style="list-style-type: none"> <li>1. System alerts the customer that no packages are currently available to view</li> <li>2. Use Case resumes on step 3</li> </ol>
<b>Exceptions:</b>	<p>2a. In step 2 of the normal flow, if the customer is not in the system, or input invalid login credentials</p> <ol style="list-style-type: none"> <li>1. System prompts user to re-enter login credentials.</li> <li>2. Customer inputs login credentials.</li> <li>3. Use Case resumes on step 2.</li> </ol>
<b>Includes:</b>	N/A
<b>Frequency of Use:</b>	20-30 times per day.
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. System is currently up. Customer has packages in the system to view.
<b>Notes and Issues:</b>	N/A

### 3.4.06 Delete Package

<b>Use Case ID:</b>	3.4.06		
<b>Use Case Name:</b>	Delete Package		
<b>Created By:</b>	Carlos Arzu	<b>Last Updated By:</b>	Ben Hanna
<b>Date Created:</b>	2018-09-13	<b>Last Revision Date:</b>	2018-11-09
<b>Actors:</b>	Primary: Marketing Manager		
<b>Description:</b>	A marketing manager or assistant need to remove a package that will no		

	longer be offered.
<b>Trigger:</b>	Marketing Manager or assistant press the delete button.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Marketing manager or assistant users need to be logged in.</li> <li>2. By clicking the delete button, Marketing Manager or assistant will be asked to enter user and password again.</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. Product/service will no longer be available.</li> </ol>
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. Marketing Manager or assistant log in...</li> <li>2. Marketing Manager or assistant choose package.</li> <li>3. Marketing manager or assistant choose, and press delete button.</li> <li>4. Marketing manager or assistant are prompt to enter user and password to delete.</li> <li>5. Package is deleted.</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>1. On step 4. if log in fails, it will ask 2 more times.</li> <li>2. If it fails, it will then block the package.</li> </ol>
<b>Exceptions:</b>	None
<b>Includes:</b>	None
<b>Frequency of Use:</b>	Used as needed.
<b>Special Requirements:</b>	None
<b>Assumptions:</b>	None
<b>Notes and Issues:</b>	None

#### 4.1.04 Create Product

<b>Use Case ID:</b>	4.1.04		
<b>Use Case Name:</b>	Create Product		
<b>Created By:</b>	Jared Greenfield	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	1. Inventory Control Employee		
<b>Description:</b>	A new product is having been approved and now must be added to the database.		
<b>Trigger:</b>	An employee clicks the add product button.		
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. Employee must be a valid Inventory Control Employee</li> <li>2. Employee must be on the view products screen</li> </ol>		

<b>Postconditions:</b>	1. The employee will be on the view products screen 2. The database has the new record added successfully
<b>Normal Flow:</b>	1. Employee clicks “add product” 2. System brings up form to fill out 3. Employee fills out form 4. Employee presses submit 5. System saves record in database 6. System pops up message saying process was successful 7. System closes form and returns employee to menu page
<b>Alternative Flows:</b>	4a. Instead of pressing submit, the employee presses the cancel button <ol style="list-style-type: none"> <li>The system pops up a message box asking if the employee is certain they want to exit</li> <li>The employee presses yes.</li> <li>The flow resumes at step 7.</li> </ol> 5a. This product has a product code that is the same as a currently active product. <ol style="list-style-type: none"> <li>The creation of the record in the database is cancelled.</li> <li>The system displays a message to the employee that the product code is the same as a currently active product.</li> <li>The employee acknowledges this.</li> <li>The flow resumes at step 3.</li> </ol>
<b>Exceptions:</b>	N/A
<b>Includes:</b>	UC-Inventory Control-View Product
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	1. What content will be on the product form?

#### 4.2.04 View Product

<b>Use Case ID:</b>	4.2.04		
<b>Use Case Name:</b>	View Product		
<b>Created By:</b>	Jared Greenfield	<b>Last Updated By:</b>	Dalton Cleveland
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-08
<b>Actors:</b>	Inventory Control Employee		
<b>Description:</b>	The employee needs to view a list of products.		
<b>Trigger:</b>	The employee clicks the button to view the product.		

<b>Preconditions:</b>	1. The assumptions are true 2. The employee is on the menu page
<b>Postconditions:</b>	1. Nothing has been changed 2. The employee is on the menu page
<b>Normal Flow:</b>	1. The employee presses the “view product” button 2. The system brings up a list of products. 3. The employee reads it 4. The employee then closes the list. 5. The System returns the employee to the menu page
<b>Alternative Flows:</b>	N/A
<b>Exceptions:</b>	1a. The system cannot find any products 1. The system alerts the user that no products could be found 2. The user acknowledges the systems prompt 3. The user is returned to the previous menu page
<b>Includes:</b>	Create, Update, Delete Product
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	None currently.
<b>Assumptions:</b>	The user is logged in  The user has access
<b>Notes and Issues:</b>	None currently.

#### 4.2.41 Browse Items

<b>Use Case ID:</b>	4.2.41		
<b>Use Case Name:</b>	Browse Items		
<b>Created By:</b>	Dani Russo/Craig Barkley	<b>Last Updated By:</b>	Craig Barkley
<b>Date Created:</b>	2018-09-21	<b>Last Revision Date:</b>	2018-11-04
<b>Actors:</b>	Primary: Customer		
<b>Description:</b>	The Customer browses items for purchase by going out to the web and access the site.		
<b>Trigger:</b>	The Customer would like to view and possibly purchase a hotel item. Selects “Product List” link and clicks through to store list.		
<b>Preconditions:</b>	1. The customer is a member or guest of a member. 2. The user has accessed the Browse Item Location.		
<b>Postconditions:</b>	1. The Customer is able view a list of items.		

	2. The Customer is able to select item to purchase.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The Customer selects “shop” and submits</li> <li>2. A list of available items to purchase is returned from database.</li> <li>3. The Database accepts request to see list item.</li> <li>4. The Program displays a list of available items (picture, description, price)</li> <li>5. Customer can choose to select items to purchase or close list.</li> </ol>
<b>Alternative Flows:</b>	<ol style="list-style-type: none"> <li>2a. In step 2 of the normal flow if there are no items to view. Then, <ol style="list-style-type: none"> <li>1. System error prompts user with purchasing information</li> <li>2. System returns user, to home screen.</li> </ol> </li> <li>4a. In step 4 of the normal flow if there are no items descriptions to view. <ol style="list-style-type: none"> <li>3. Then there should be a placeholder with alt image information.</li> <li>4. System returns user, to view items page.</li> </ol> </li> </ol>
<b>Exceptions:</b>	<ol style="list-style-type: none"> <li>3a. In step 3 of the normal flow, if the customer enters and invalid Search Request <ol style="list-style-type: none"> <li>1. Transaction is disapproved</li> <li>2. Message to user to search for another product.</li> <li>3. Customer enter in information for a new search.</li> </ol> </li> </ol>
<b>Includes:</b>	2. Personnel could also email Resort to check on out of stock items. The Manager would also have access to the list and could help any resort guest in person with any questions on requisitioning items or products that are on the resort list.
<b>Frequency of Use:</b>	This Use Case will be executed for Each item list that is searched by members of the Resort, both while staying at the resort and as an external customer through the resort user facing online portal.
<b>Special Requirements:</b>	The viewer of the records is a resort member or guest.
<b>Assumptions:</b>	User is logged in. User has permissions. From this view the Customer can select an item.
<b>Notes and Issues:</b>	<ol style="list-style-type: none"> <li>1. They have corrected permission to view item details.</li> </ol>

#### 4.3.05 Edit Product

<b>Use Case ID:</b>	4.3.05		
<b>Use Case Name:</b>	Edit Product		
<b>Created By:</b>	Jared Greenfield	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	<ol style="list-style-type: none"> <li>1. Inventory Control Employee</li> </ol>		
<b>Description:</b>	The employee needs to change the description or some other attribute		

	about a certain product.
<b>Trigger:</b>	The employee clicks the button to edit the product.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The employee is a valid Inventory Control Employee</li> <li>2. The employee is on the view product details page</li> <li>3. There is a product record created that can be edited</li> </ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"> <li>1. The employee is on the product details page</li> <li>2. The database has been changed to reflect this new data</li> </ol>
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The employee presses the “edit product” button</li> <li>2. The system brings up the filled in form.</li> <li>3. The employee edits the needed section</li> <li>4. The employee then submits the form</li> <li>5. The system then brings up a message box saying that the change was successful.</li> <li>6. The System returns the employee to the menu page</li> </ol>
<b>Alternative Flows:</b>  [Alternative Flow 1 – Not in Network]	<p>4a. The employee presses the cancel button instead of the submit button</p> <ol style="list-style-type: none"> <li>1. The system brings up a message box asking if the employee really wants to do that.</li> <li>2. The employee presses yes.</li> <li>3. The system does not save the changes and the record remains the same.</li> <li>4. Flow resumes at step 6.</li> </ol> <p>4b. The employee presses the cancel button instead of the submit button</p> <ol style="list-style-type: none"> <li>1. The system brings up a message box asking if the employee really wants to do that.</li> <li>2. The employee presses no.</li> <li>3. The message box closes</li> <li>4. Flow resumes at step 3.</li> </ol>
<b>Exceptions:</b>	N/A
<b>Includes:</b>	UC-Inventory Control-View Product Details
<b>Frequency of Use:</b>	On demand
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	N/A
<b>Notes and Issues:</b>	N/A

#### 4.4.02 Deactivate Product

<b>Use Case ID:</b>	4.4.02
---------------------	--------

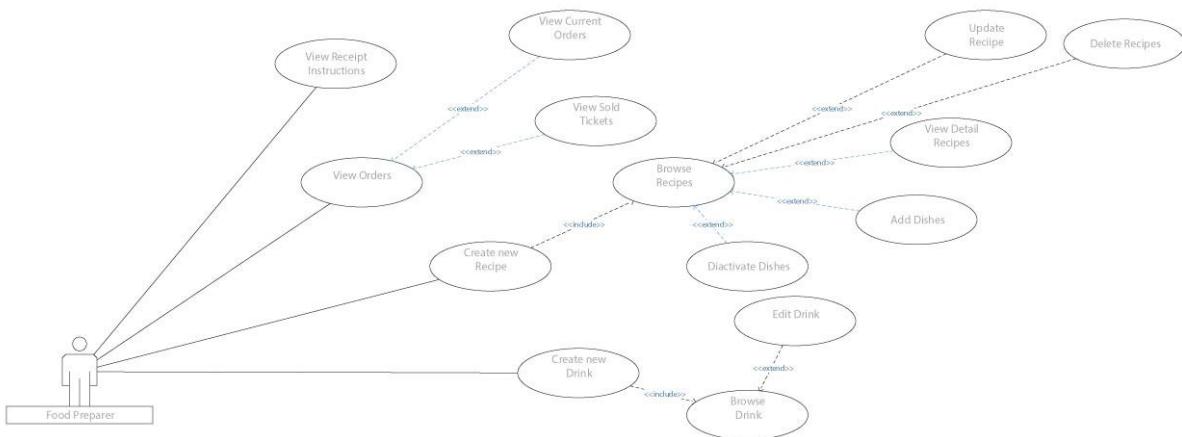
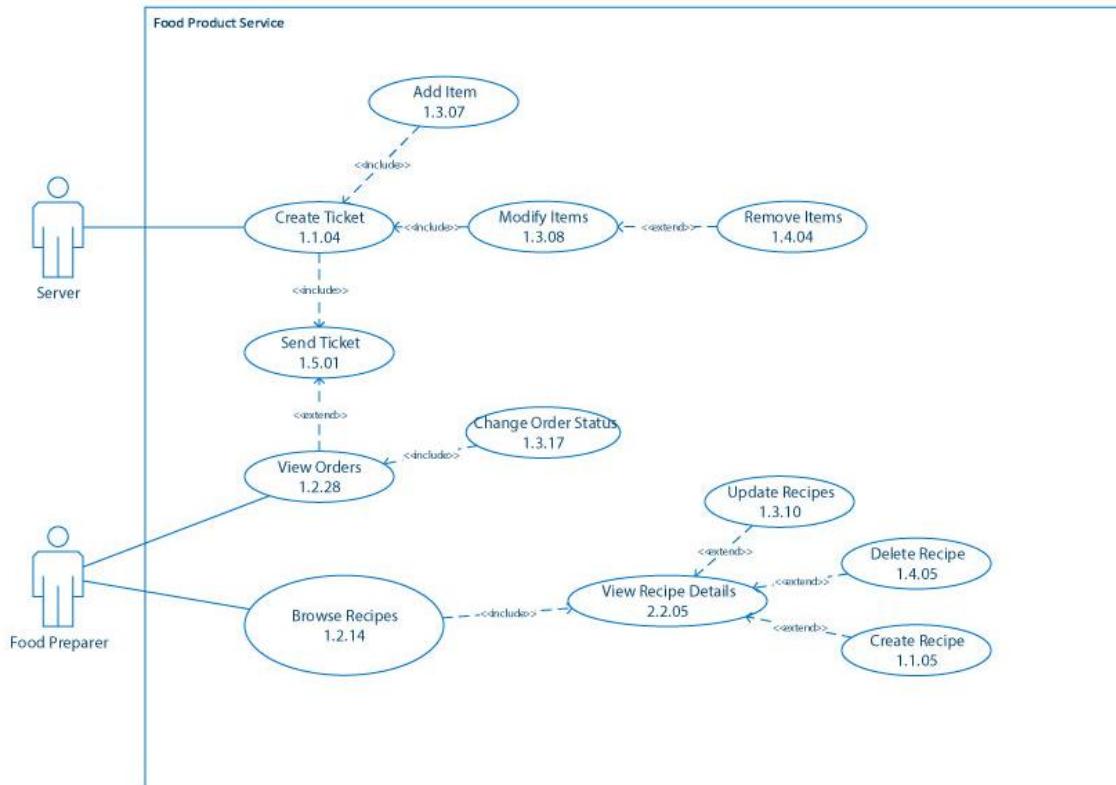
<b>Use Case Name:</b>	Deactivate Product		
<b>Created By:</b>	Jared Greenfield	<b>Last Updated By:</b>	Jeff Stobb
<b>Date Created:</b>	2018-09-12	<b>Last Revision Date:</b>	2018-11-01
<b>Actors:</b>	1. Inventory Control Employee		
<b>Description:</b>	The employee needs to remove a product from active status because the company no longer makes it.		
<b>Trigger:</b>	The employee clicks the button to deactivate the product.		
<b>Preconditions:</b>	1. The employee is a valid Inventory Control Employee 2. The employee is on the view product details page 3. There is a product record created that can be deactivated		
<b>Postconditions:</b>	1. The employee is on the product details page 2. The database has been changed to reflect this new data		
<b>Normal Flow:</b>	1. The employee presses the “edit product” button 2. The system brings up the filled in form. 3. The employee edits the needed section 4. The employee then submits the form 5. The system commits the change to the database 6. The system then brings up a message box saying that the change was successful. 7. The System returns the employee to the product details page		
<b>Alternative Flows:</b>	4a. The employee presses the cancel button instead of the submit button		
<b>[Alternative Flow 1 – Not in Network]</b>	1. The system brings up a message box asking if the employee really wants to do that. 2. The employee presses yes. 3. The system does not save the changes and the record remains the same. 4. Flow resumes at step 6.		
	4b. The employee presses the cancel button instead of the submit button		
	1. The system brings up a message box asking if the employee really wants to do that. 2. The employee presses no. 3. The message box closes 4. Flow resumes at step 3.		
<b>Exceptions:</b>	5a. The record cannot be found to deactivate. 1. The update is cancelled, and the system sends a message to the employee that the update has failed. 2. The employee acknowledges this message.		
<b>Includes:</b>	UC-Inventory Control-ViewProductDetails		
<b>Frequency of Use:</b>	On demand		

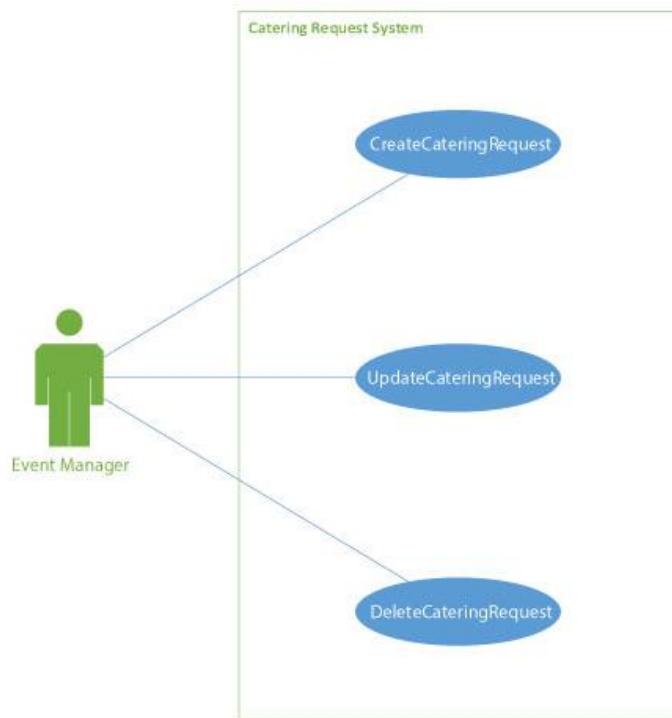
<b>Special Requirements:</b>	N/A
<b>Assumptions:</b>	User is logged in. User has permissions.
<b>Notes and Issues:</b>	N/A

## Use Case Diagrams

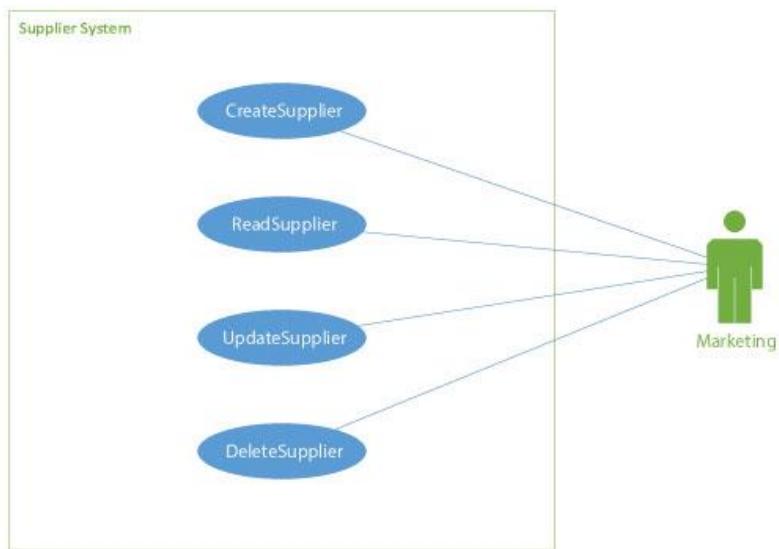
The following Use Case Diagrams are simple draws separated by subsystems with in the system. Each diagram will show what use case an actor will interact with and how other use cases relate to that use case.

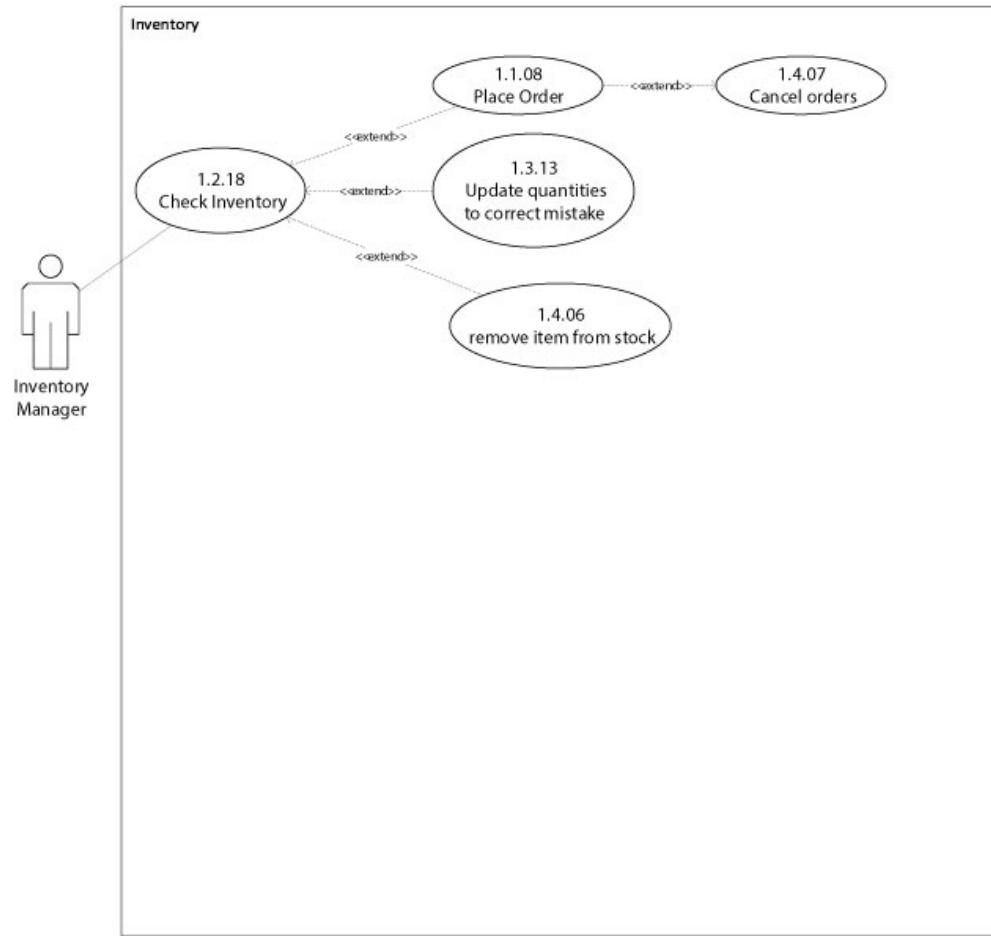
### Food Operations



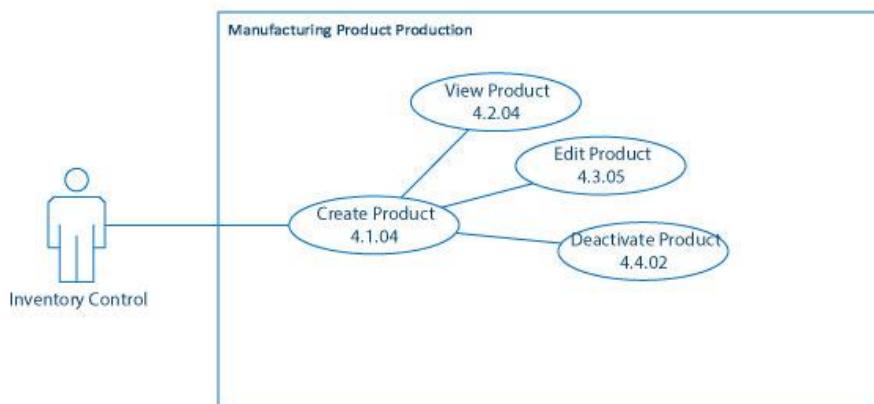


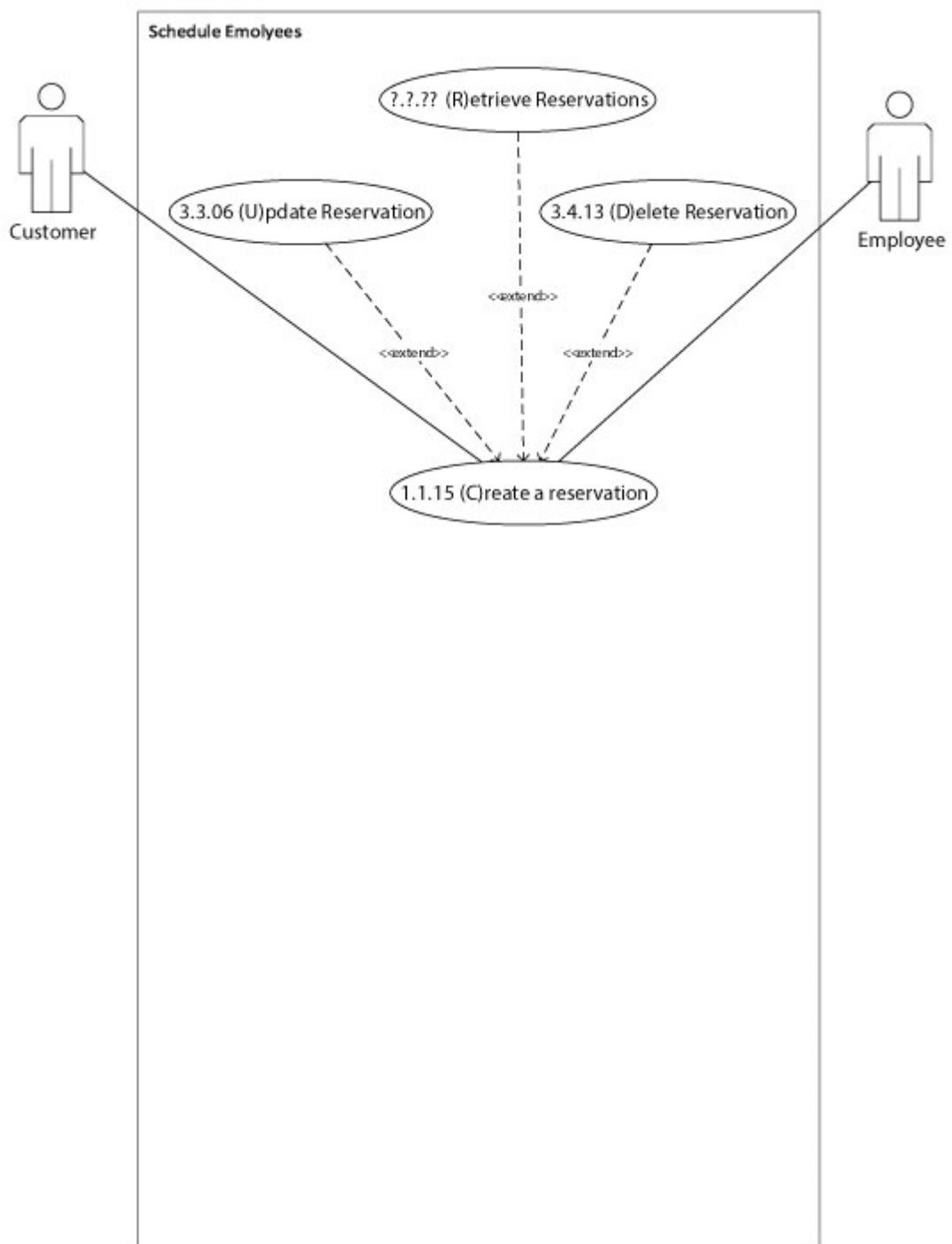
## Supply Operations



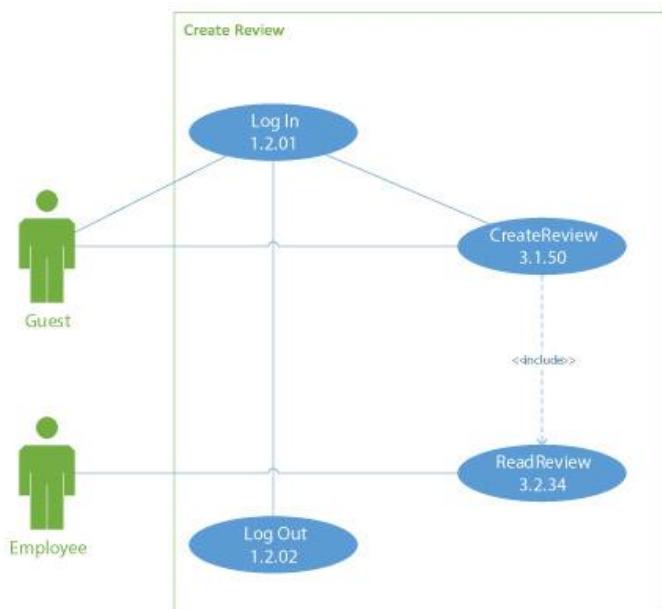
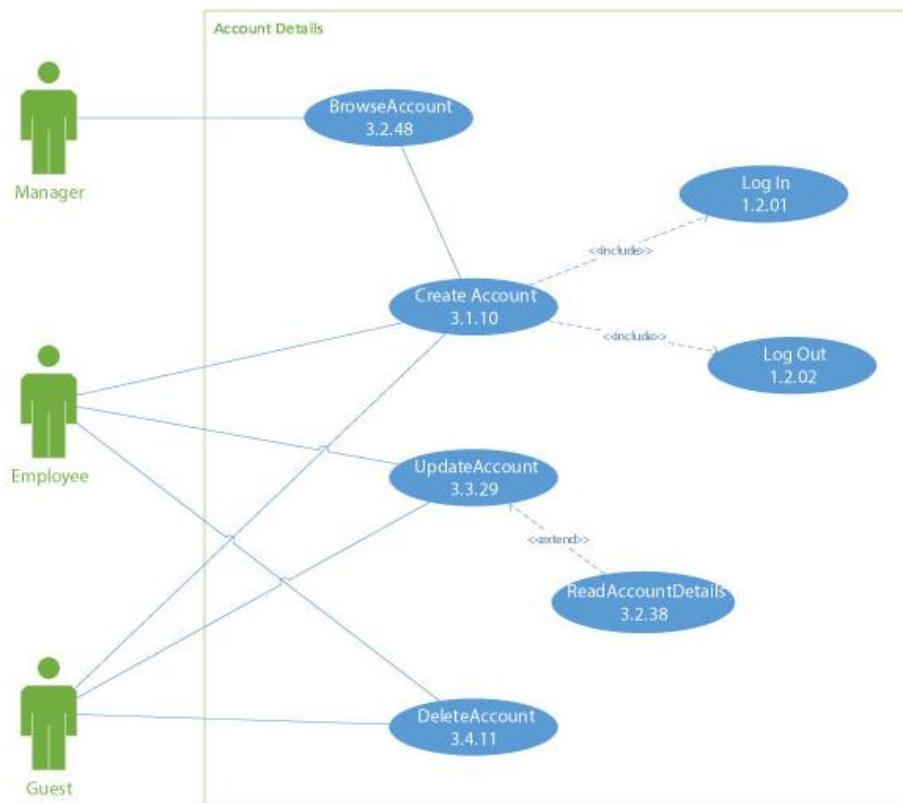


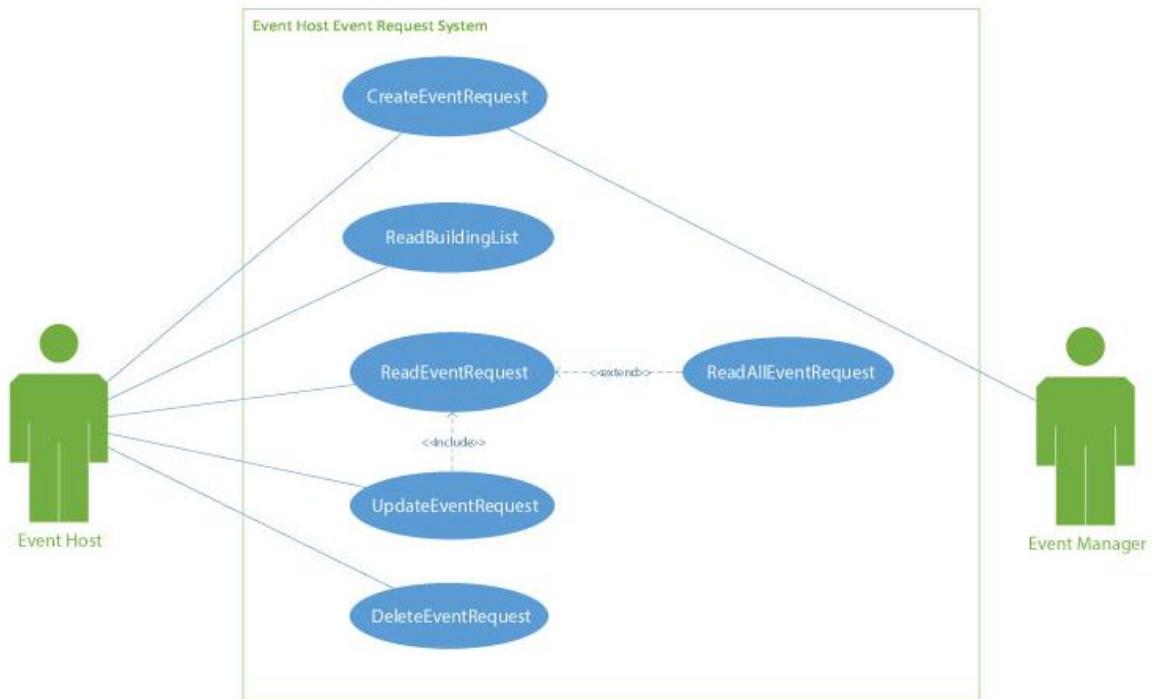
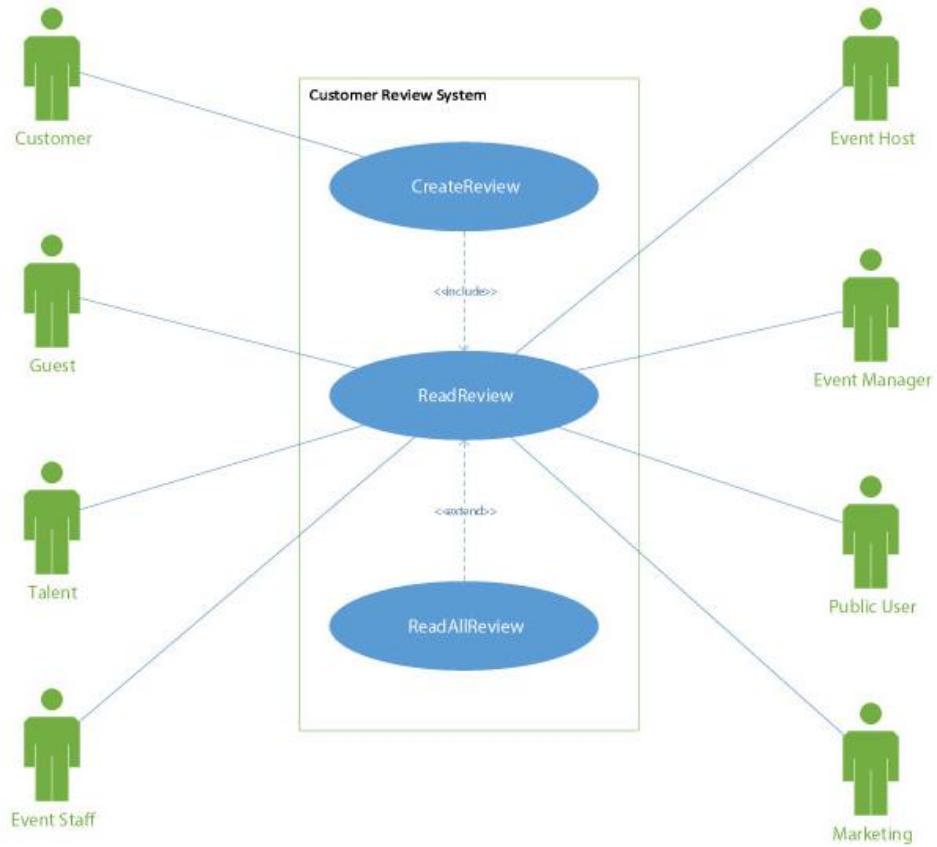
## Facility Operations

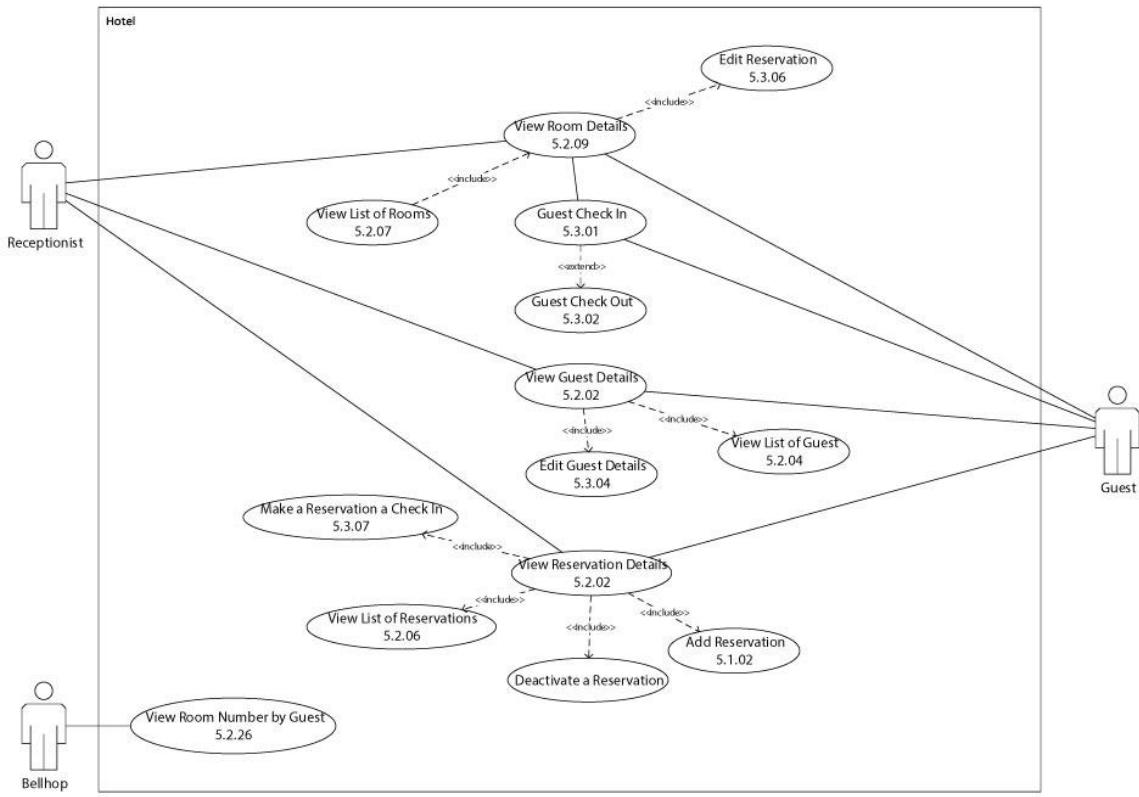
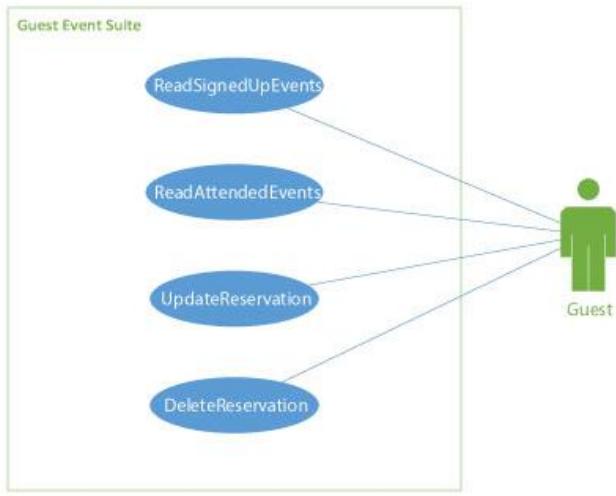




## Guest Operations

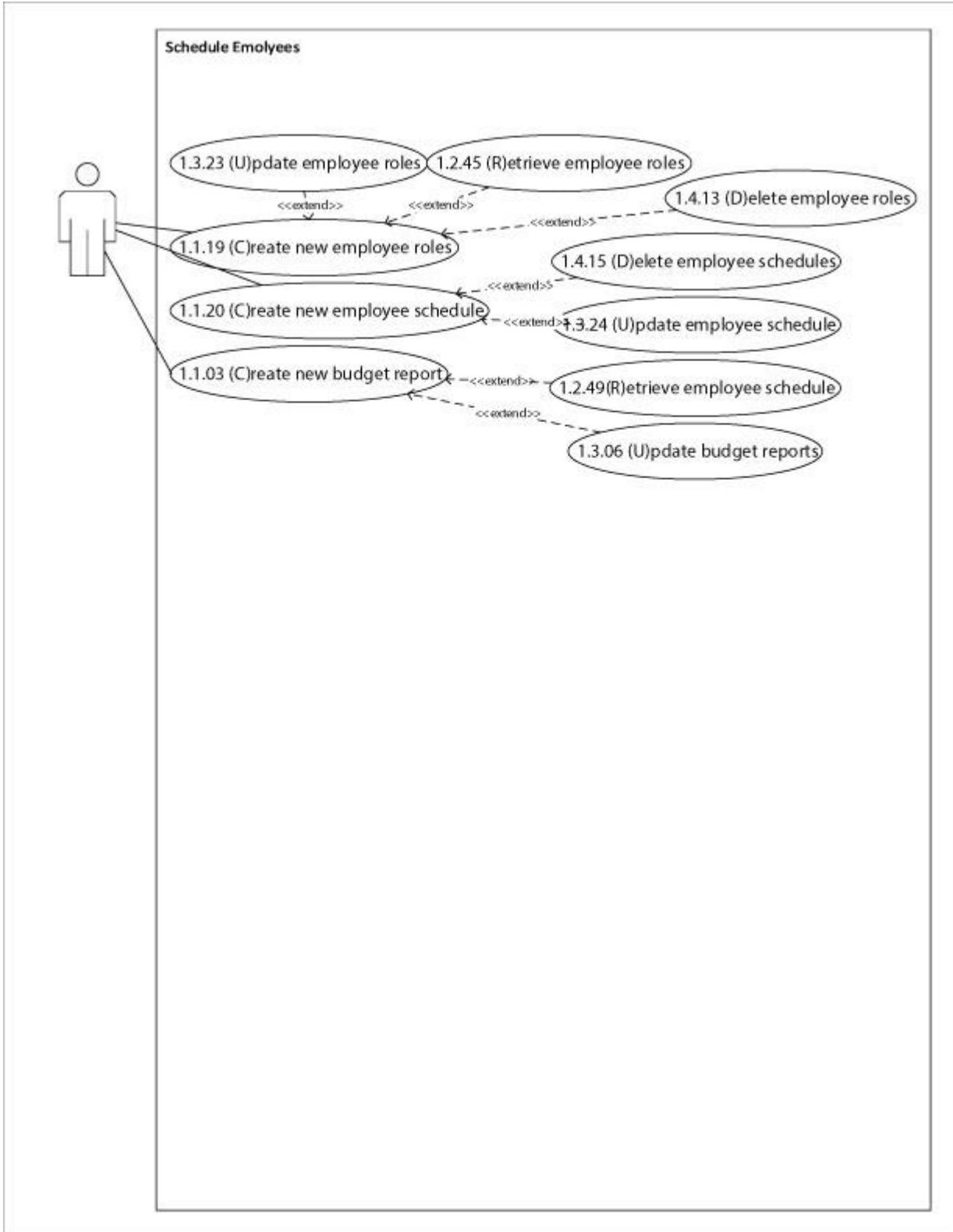




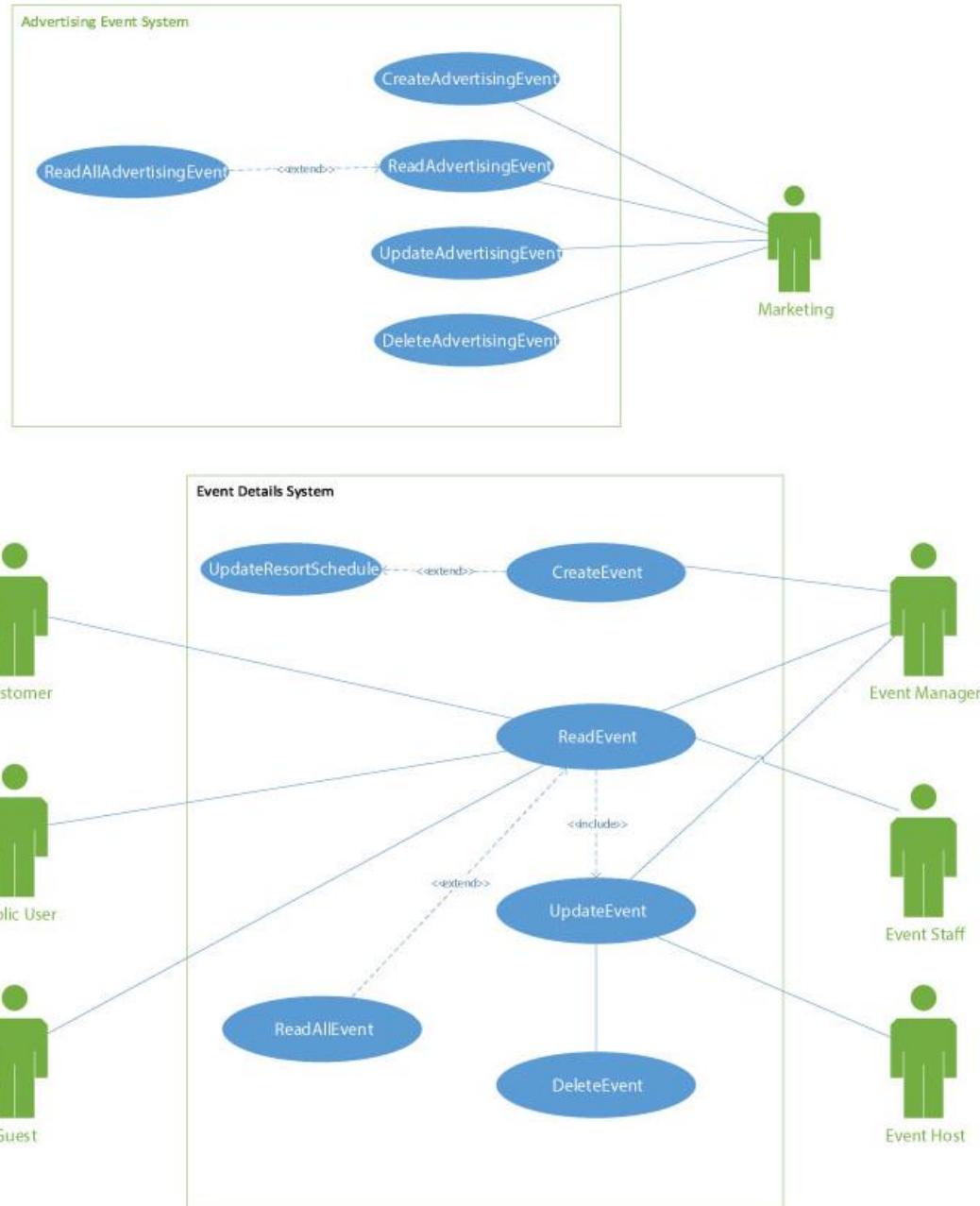


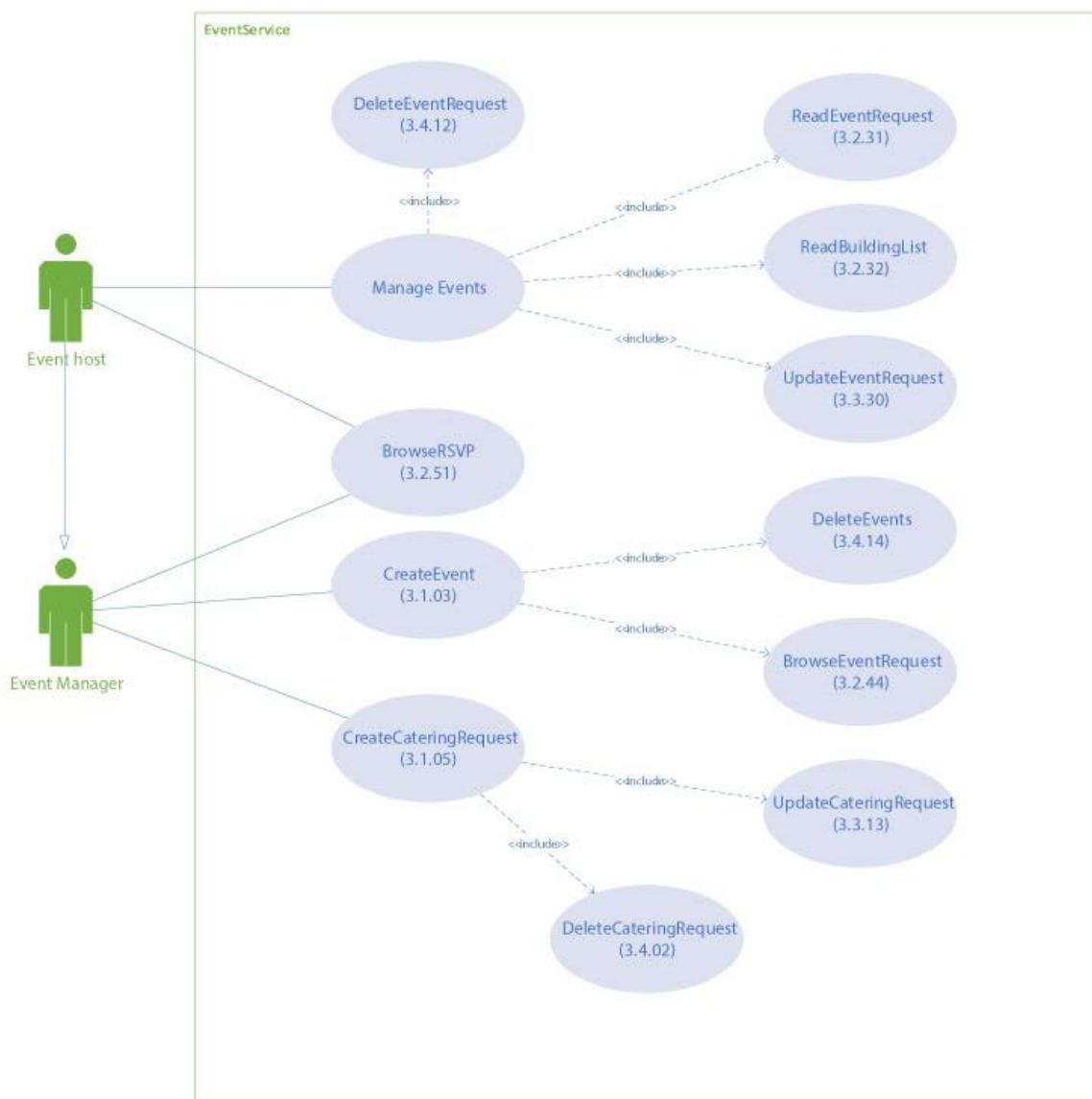
## Personnel Management

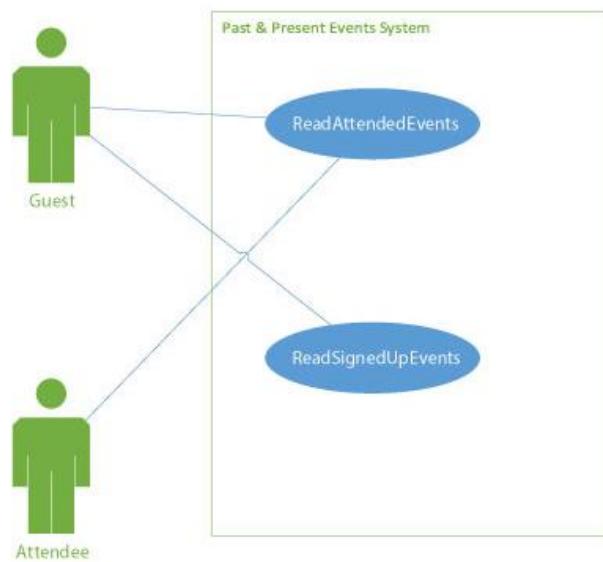
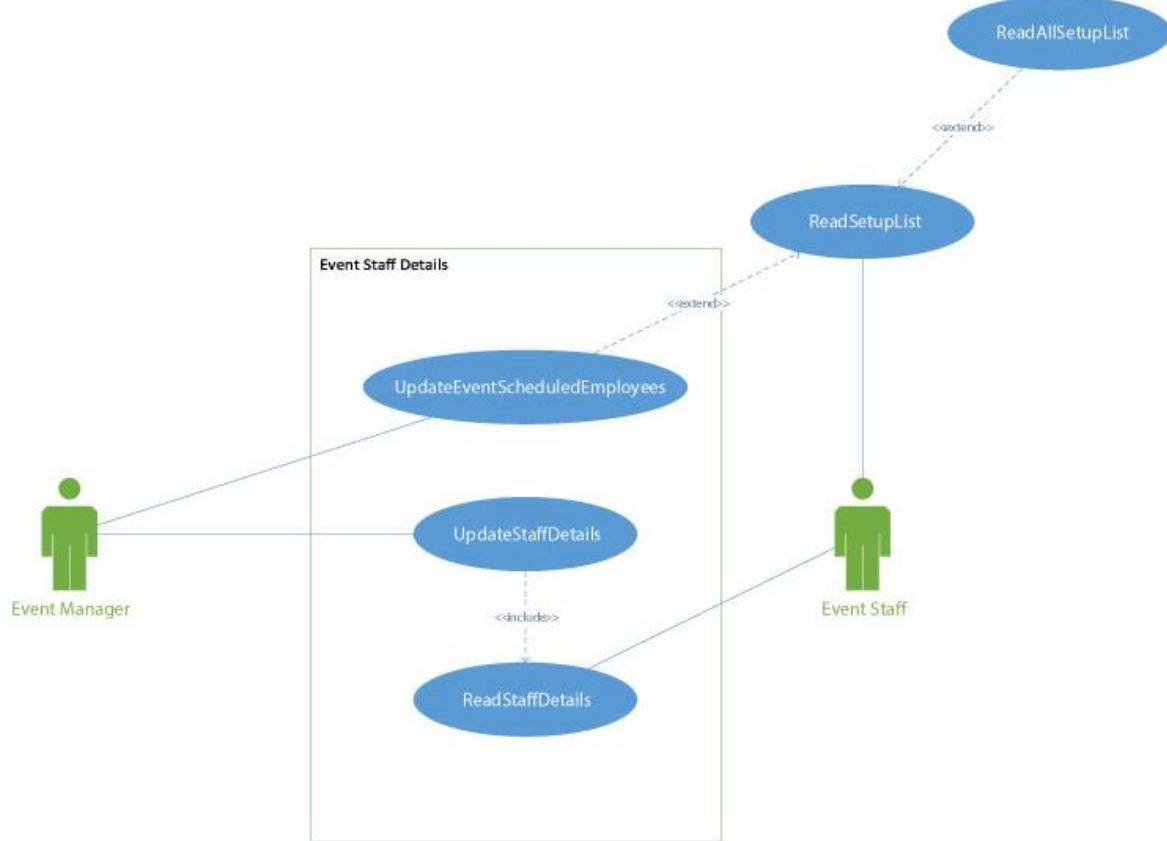


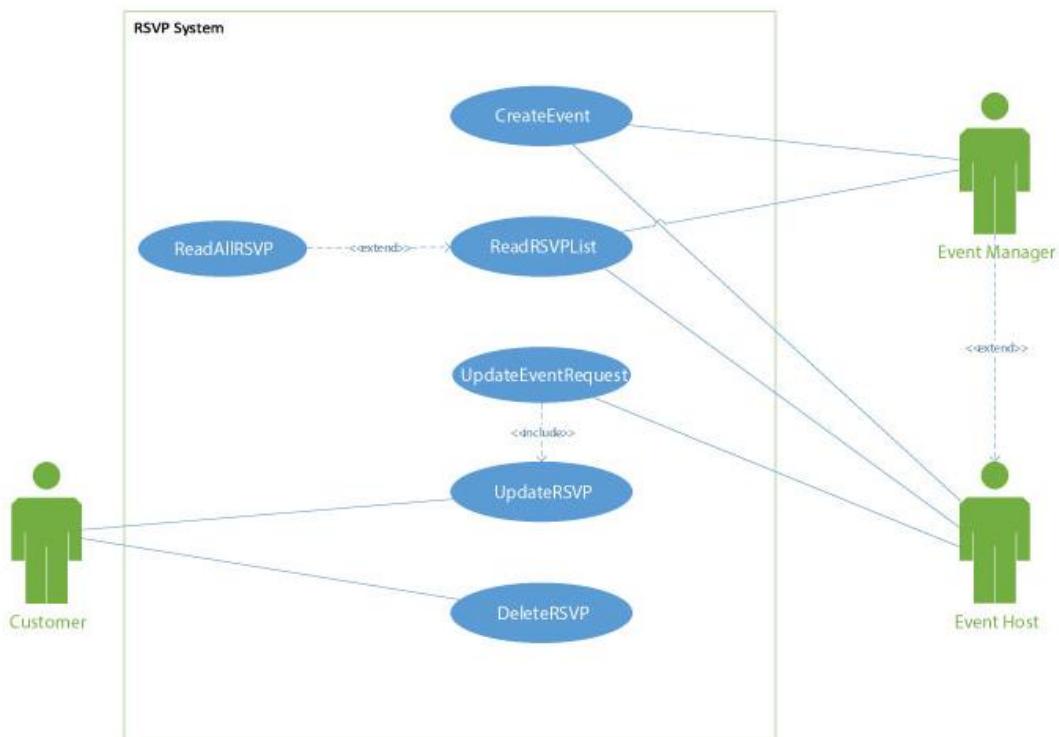
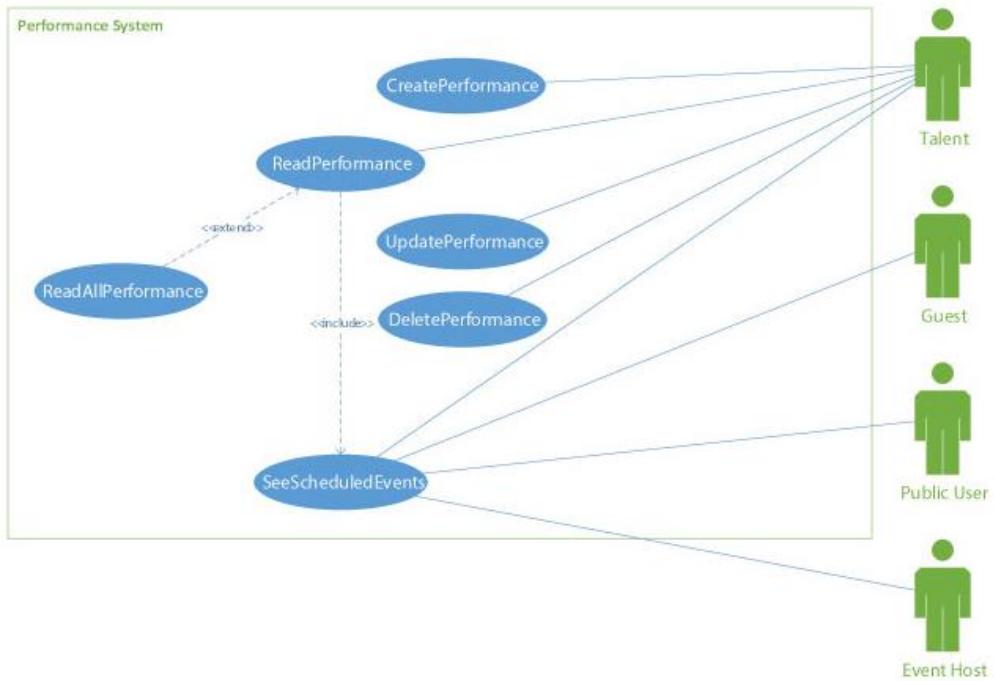


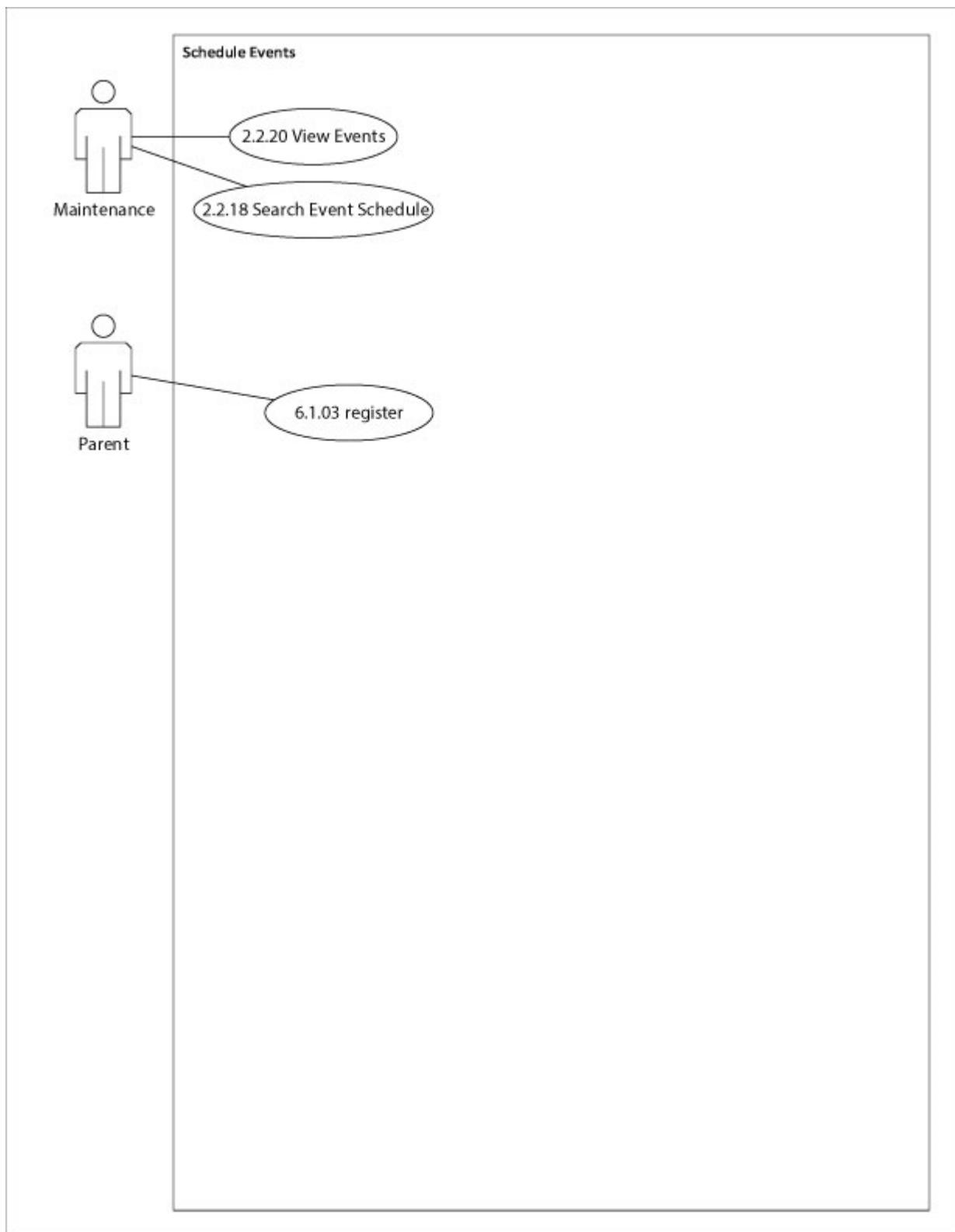
## Event Operations

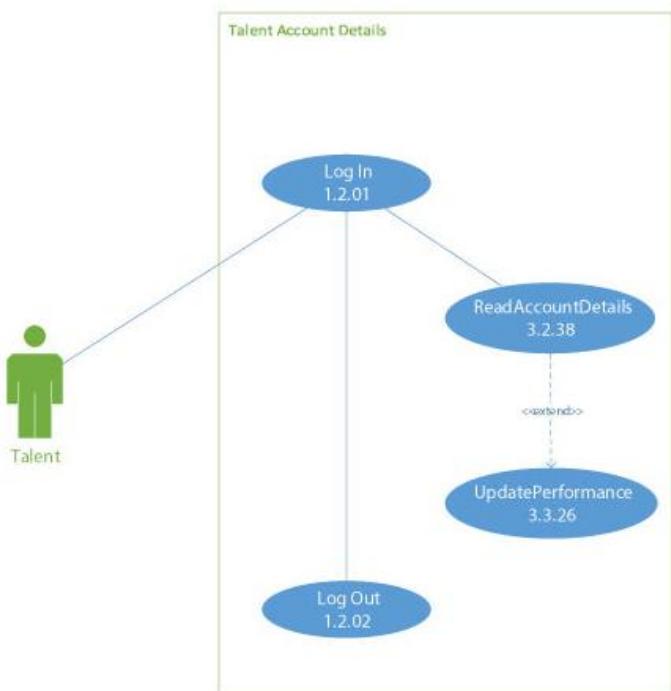
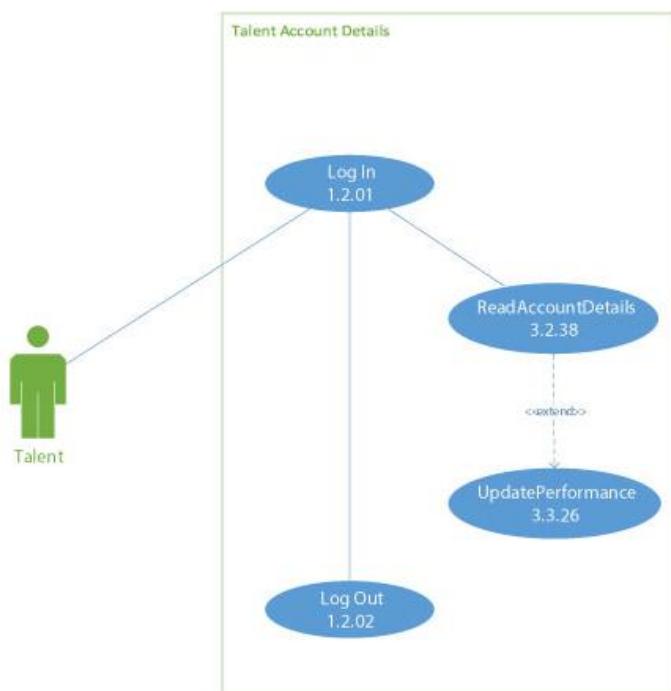




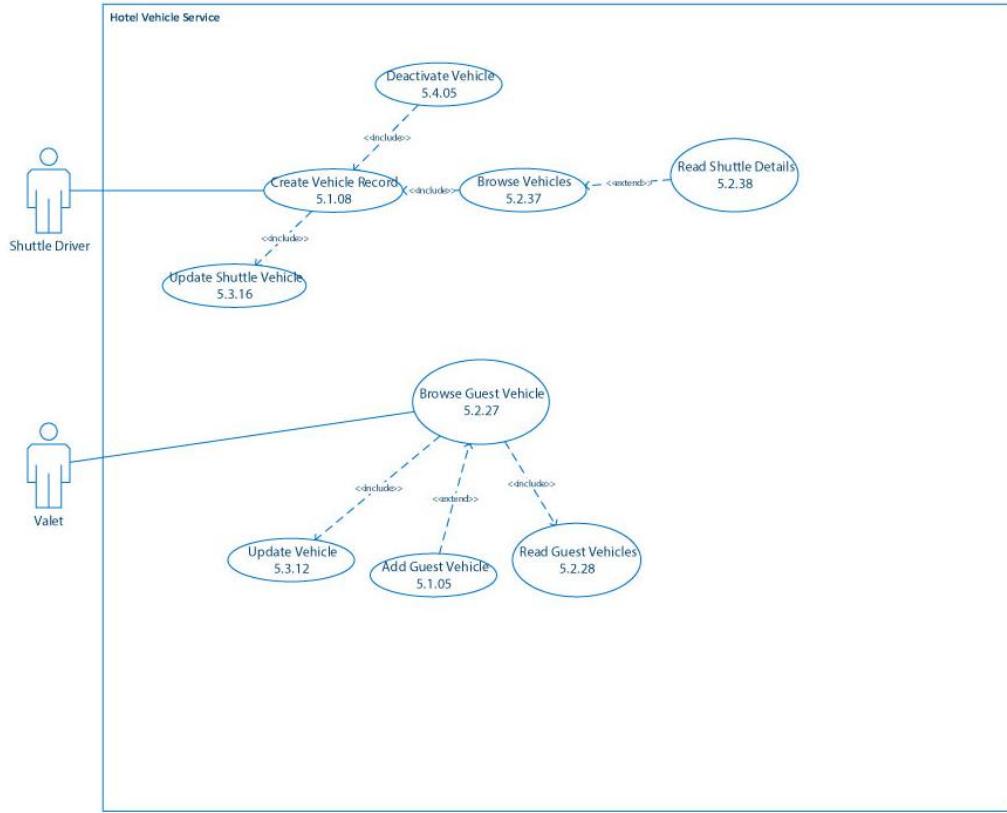


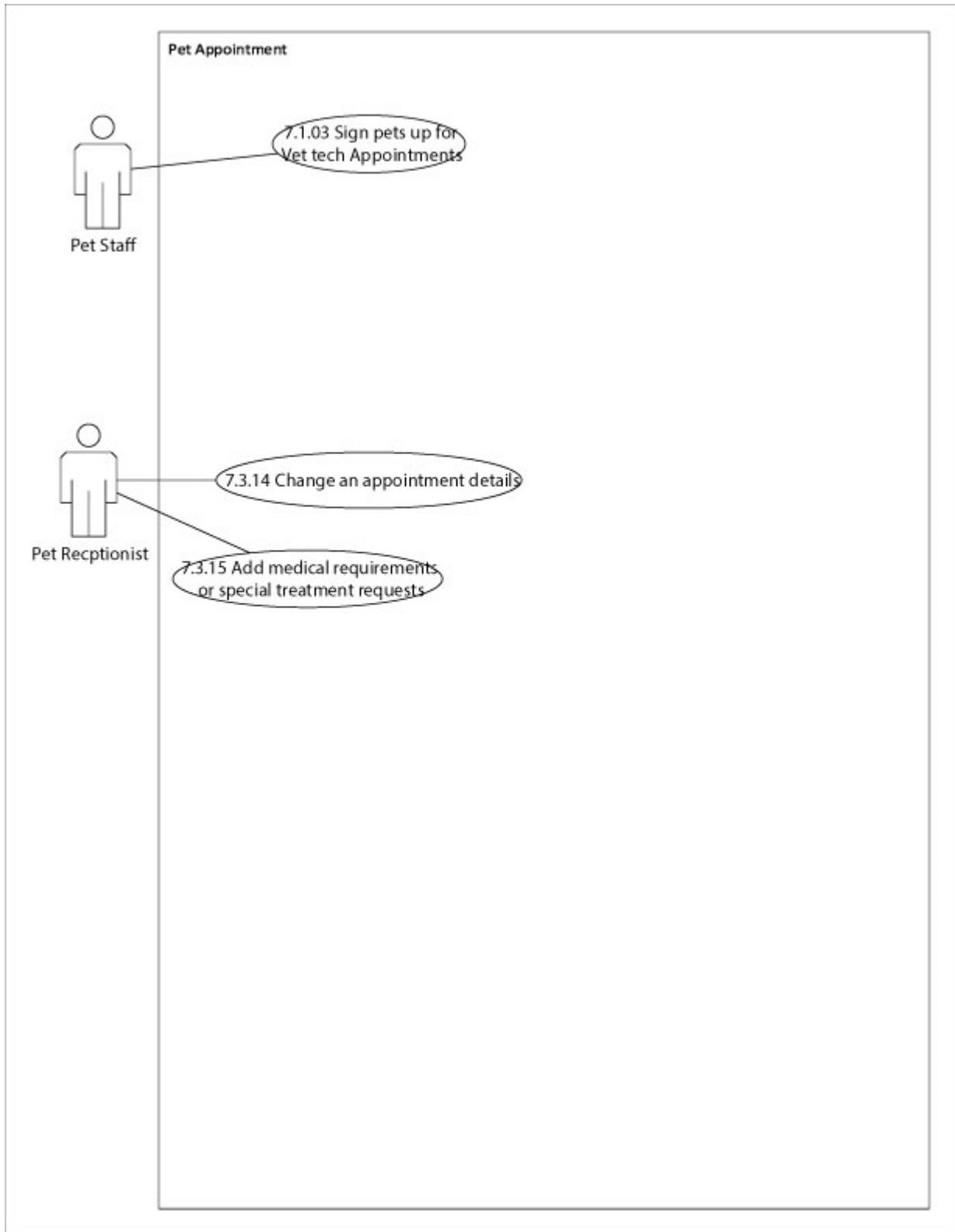


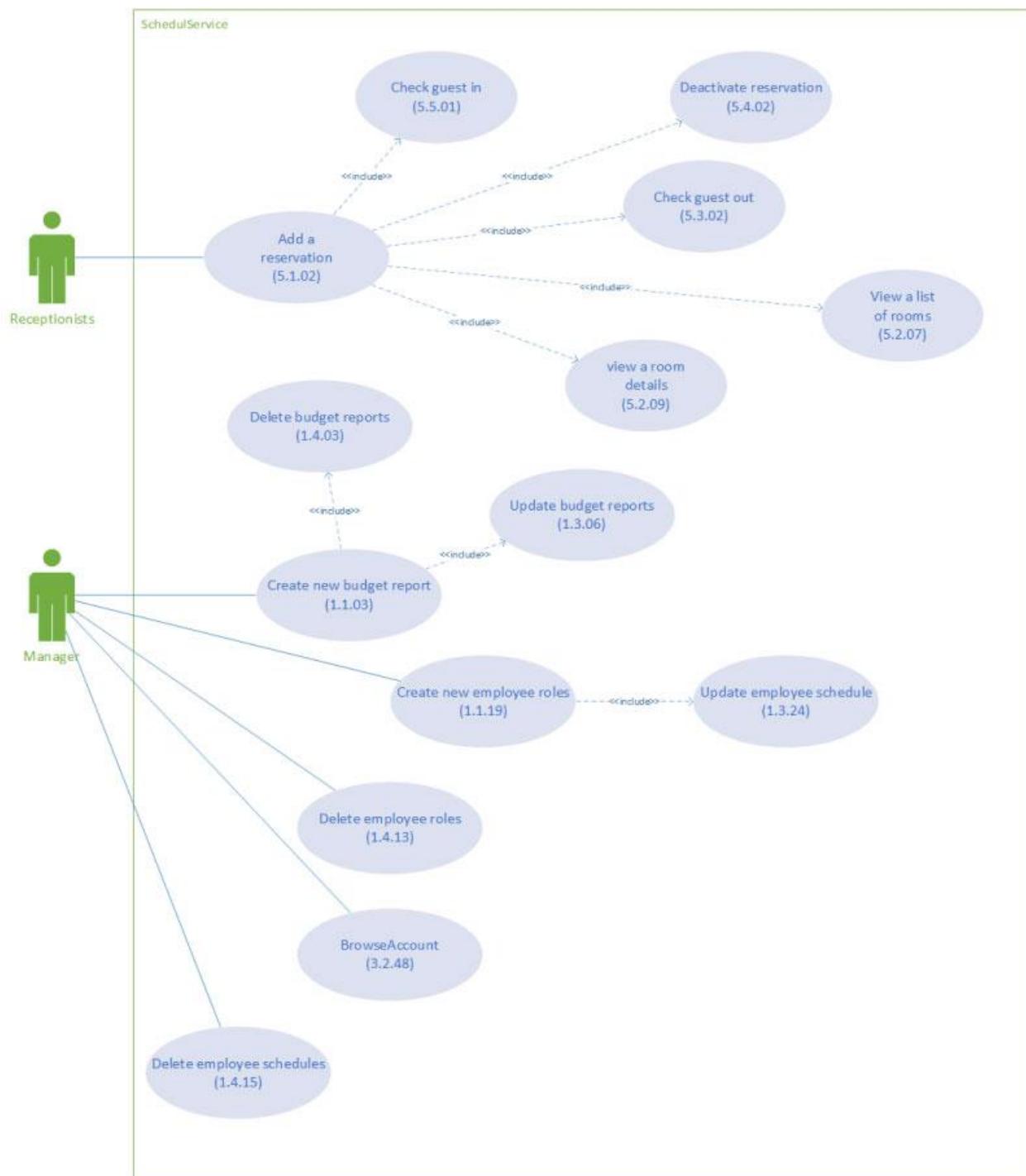


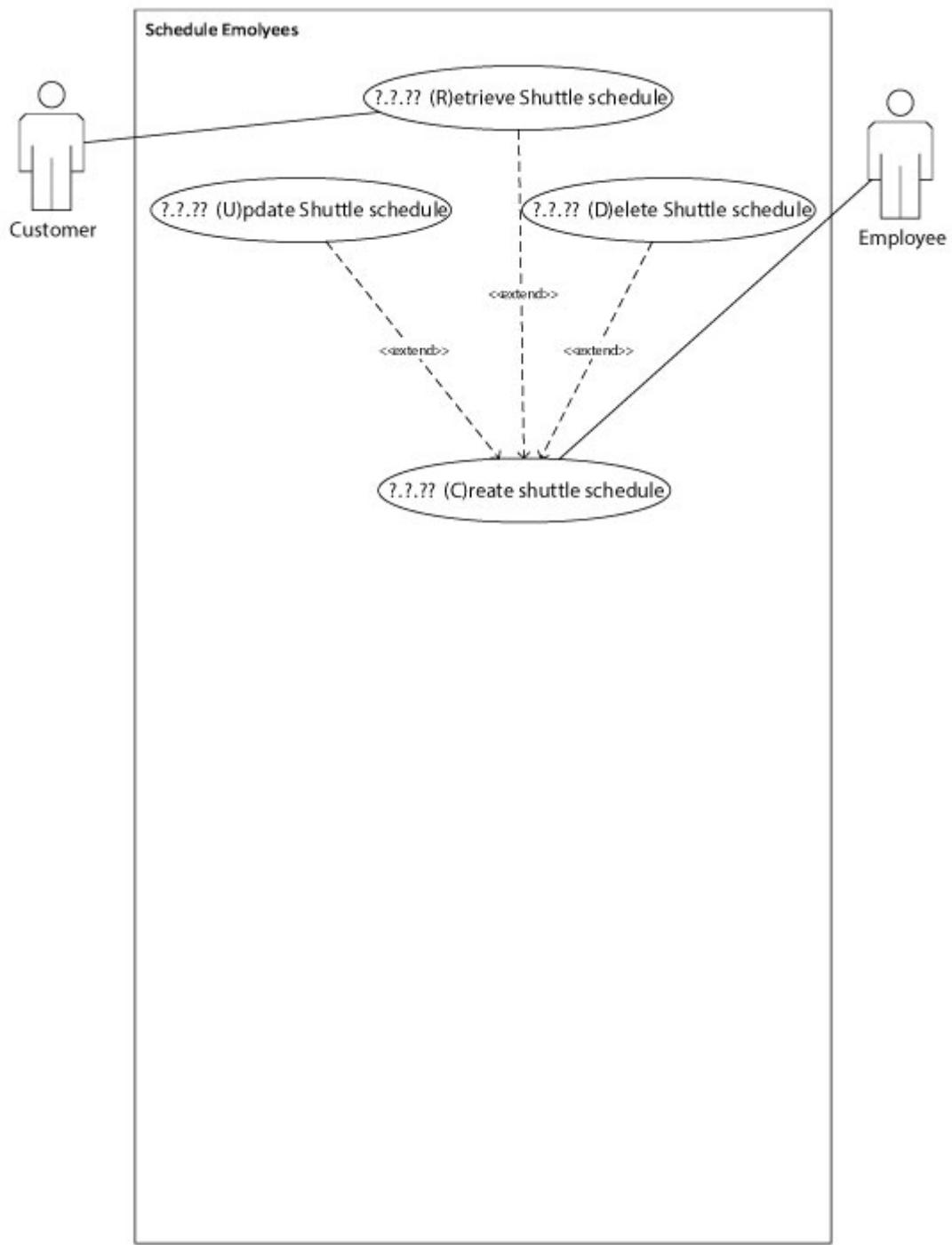


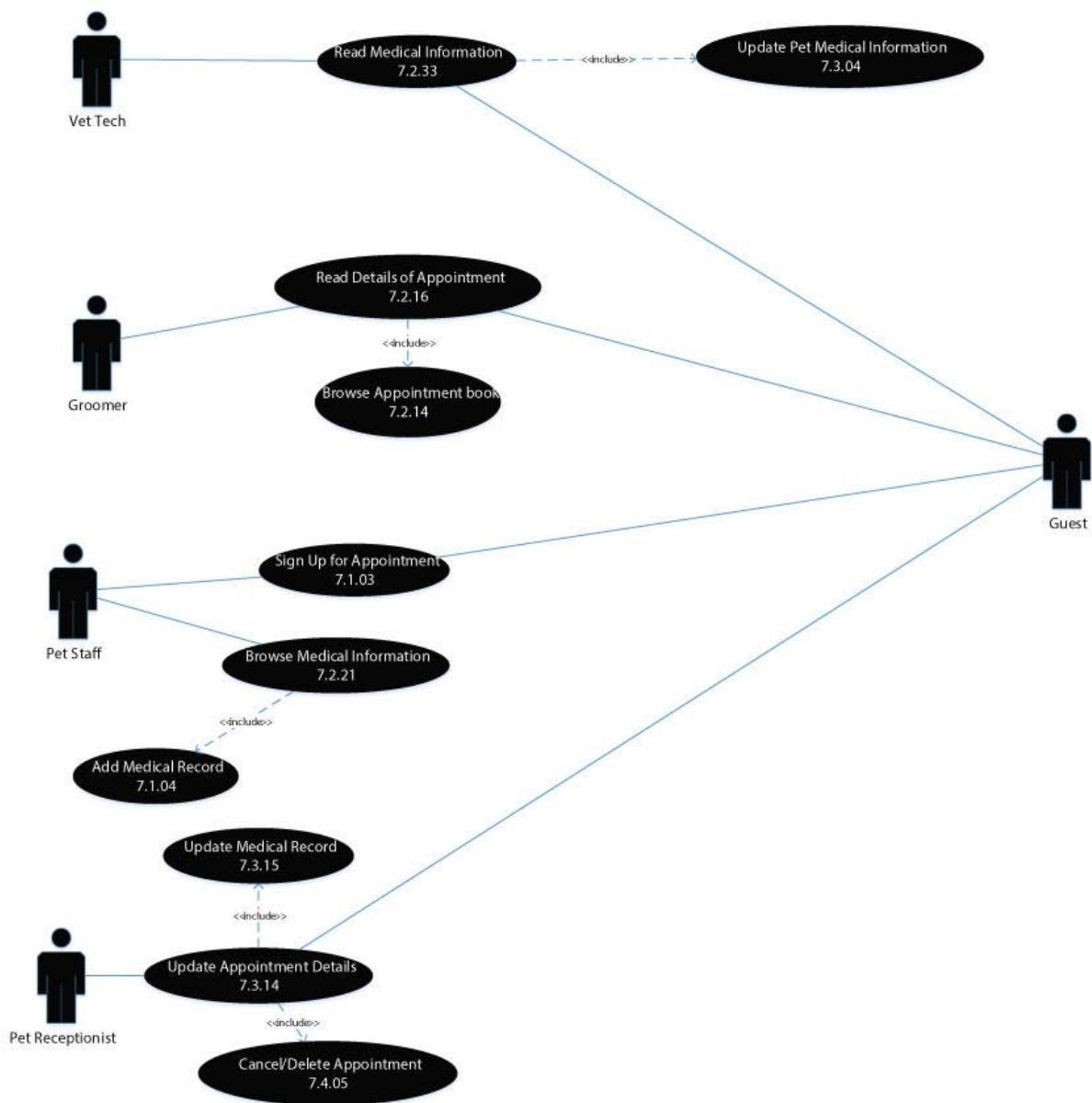
## Service and Appointment



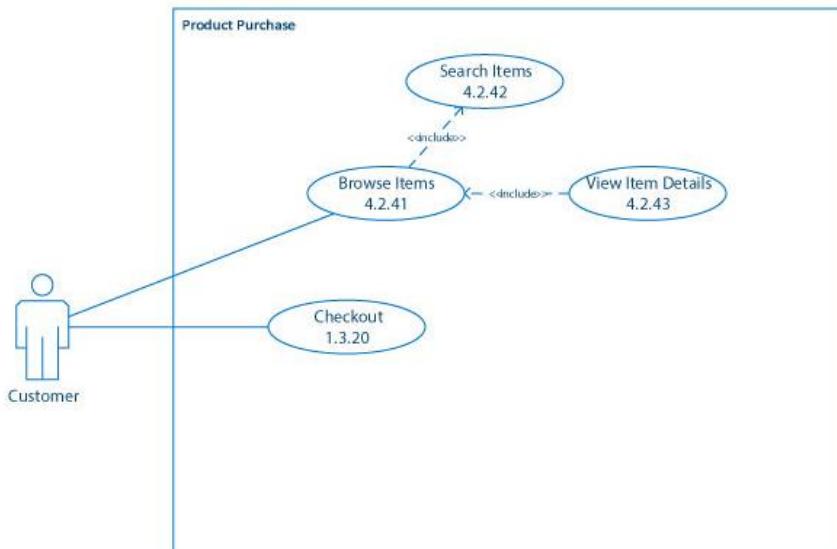
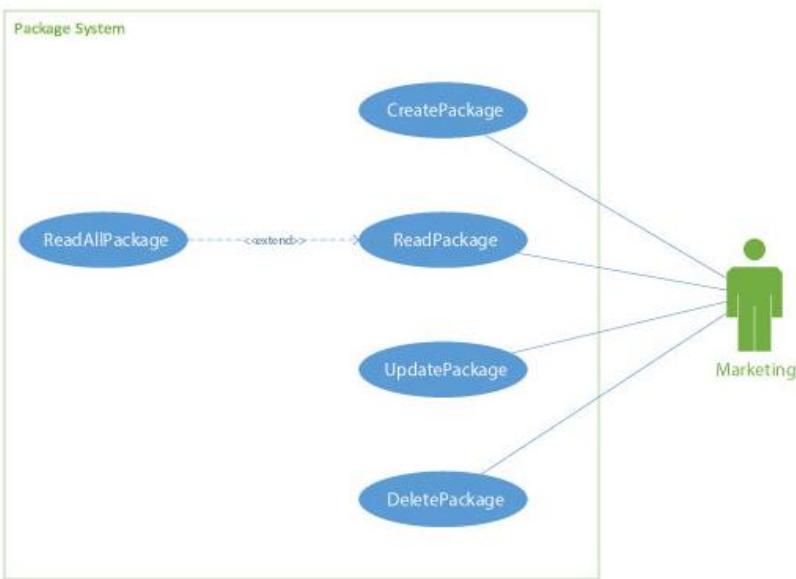


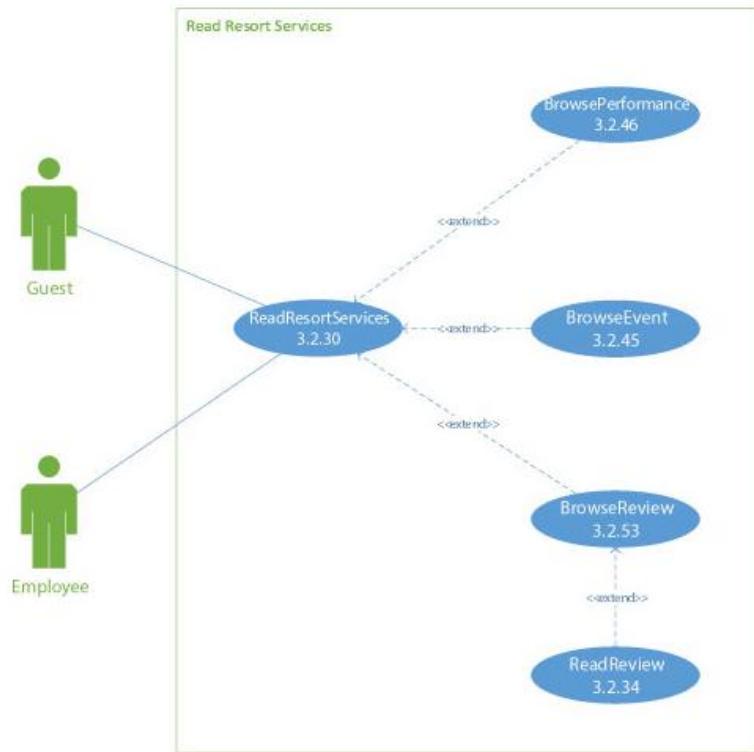






## Product Operations



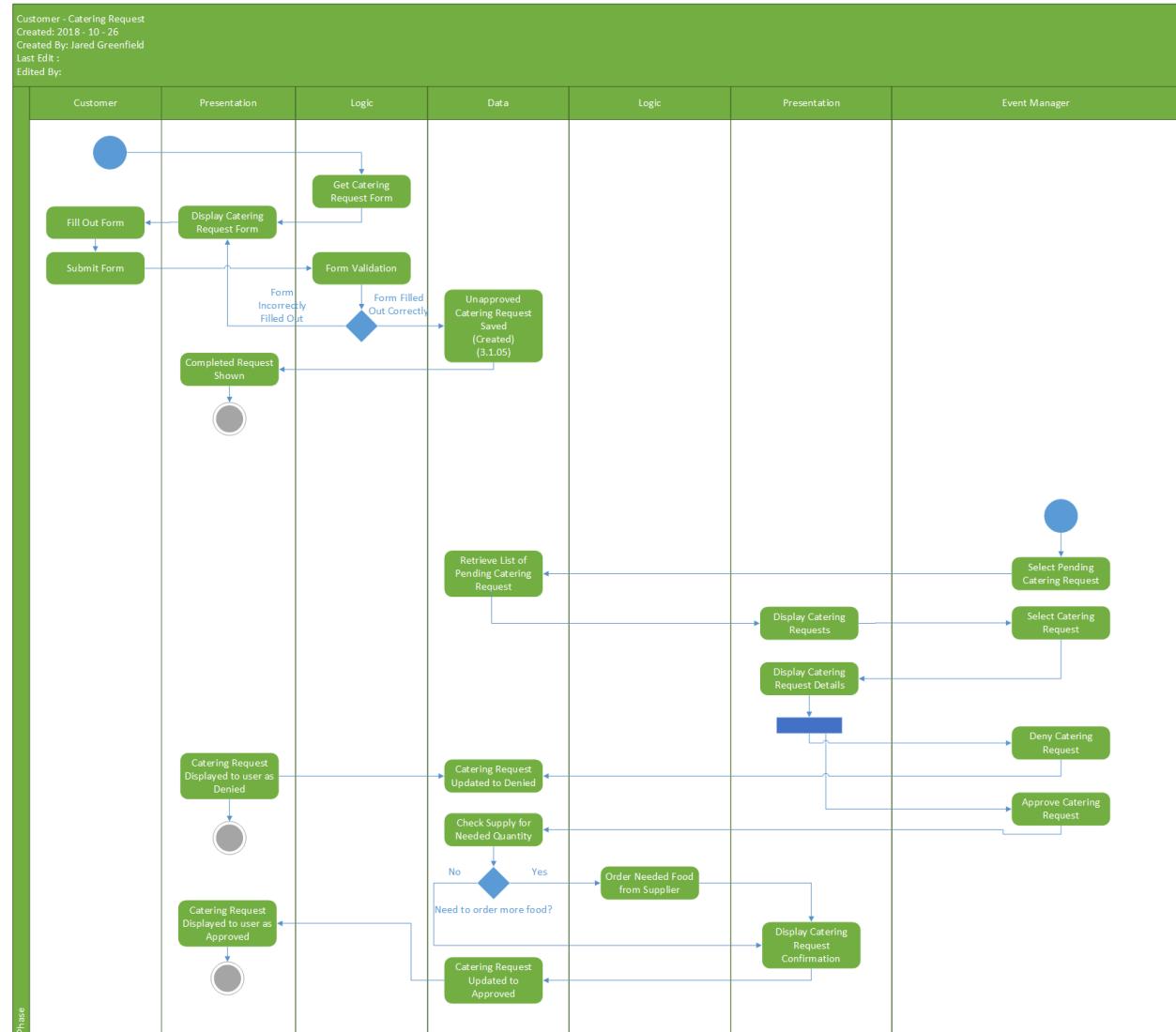


## Activity Diagrams

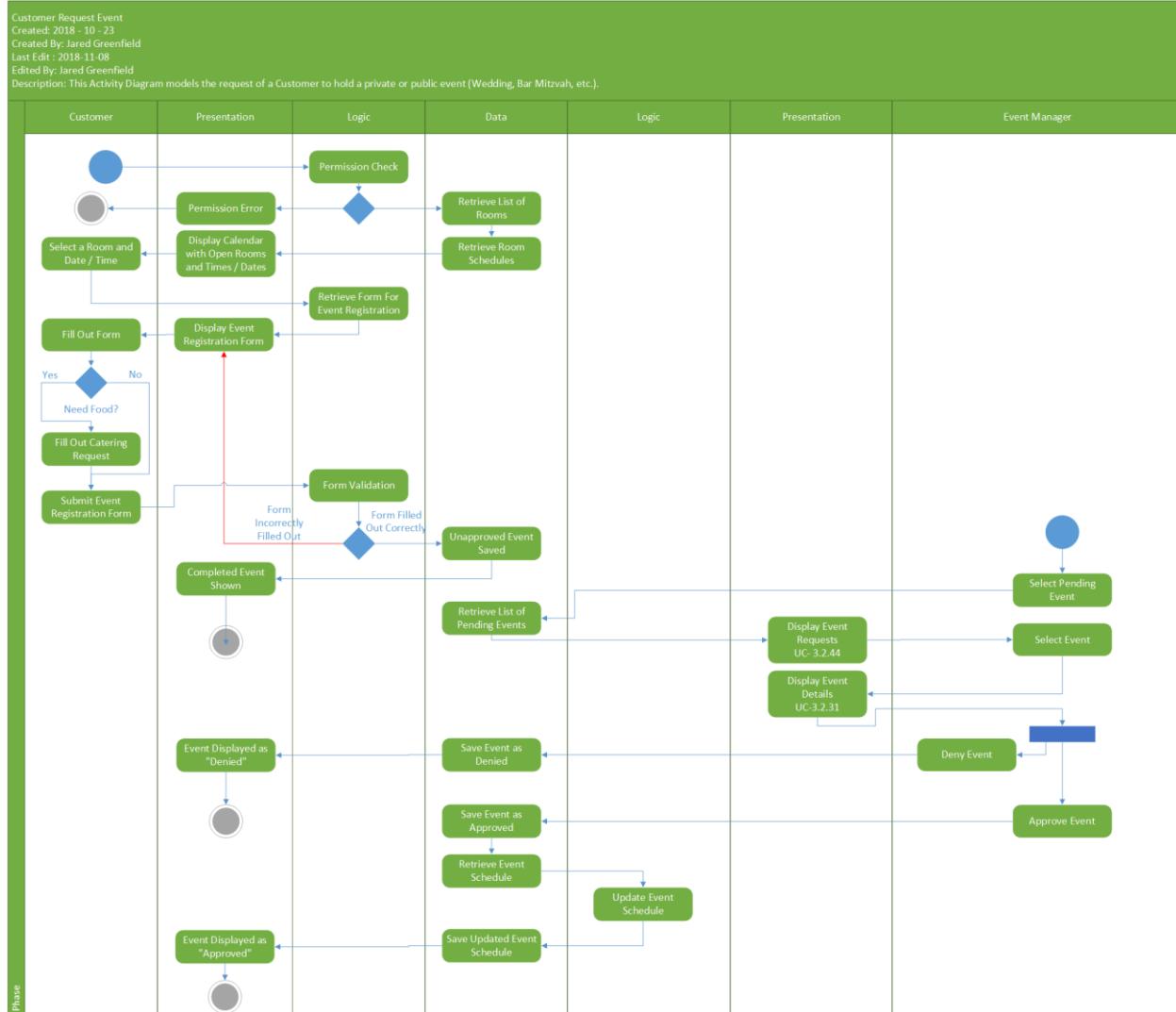
### Customer Operations

The following activity diagrams detail various Customer interactions and their progression within the resort such as purchasing goods and services, reserving rooms, and utilizing amenities. Customer requests that require approval or authorization are transferred to the appropriate resort Employee who can complete the transaction.

#### Catering Request



## Request Event



## Reserve Appointment

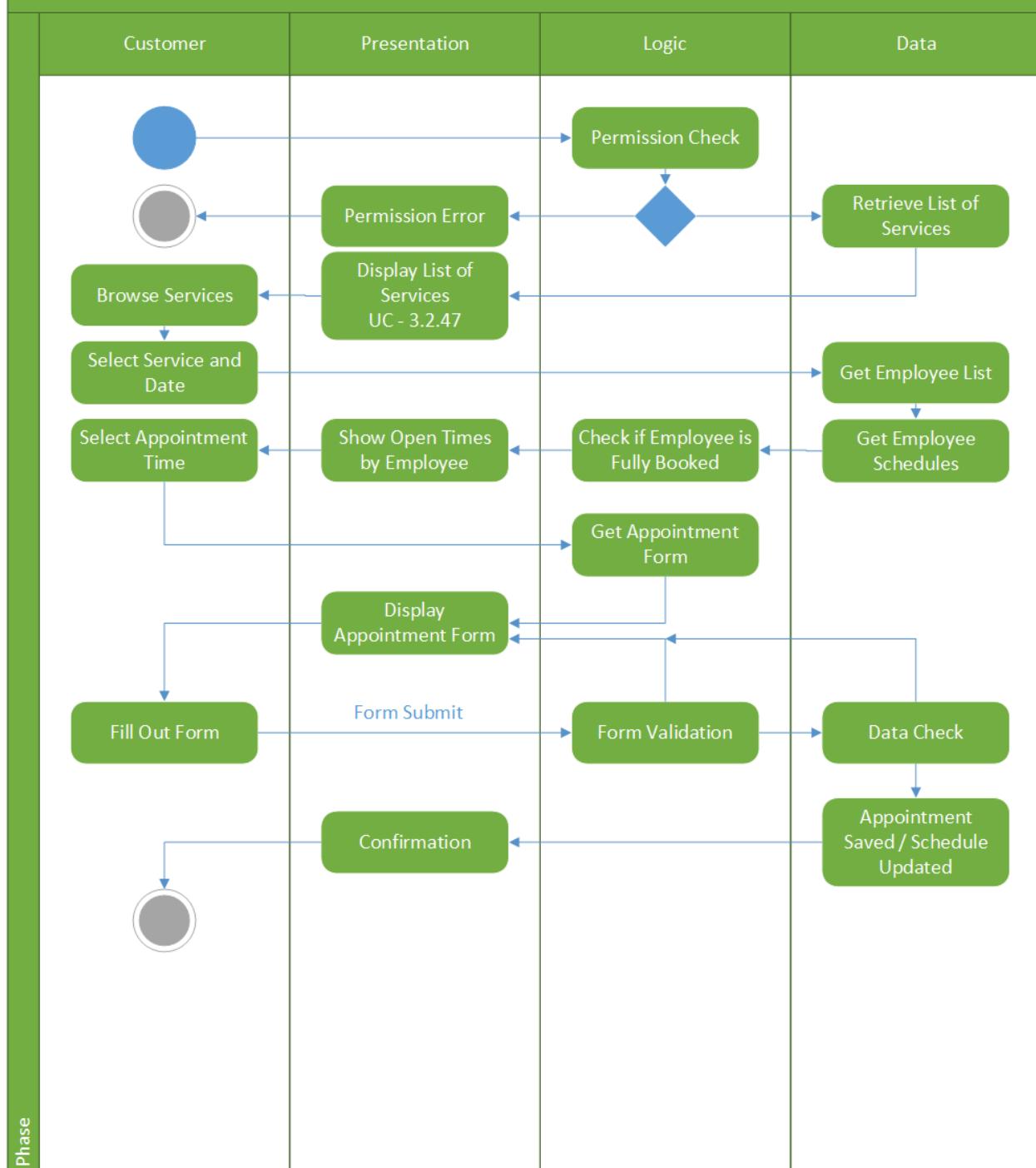
Customer Reserve Appointment By Employee

Created: 2018 - 10 - 17

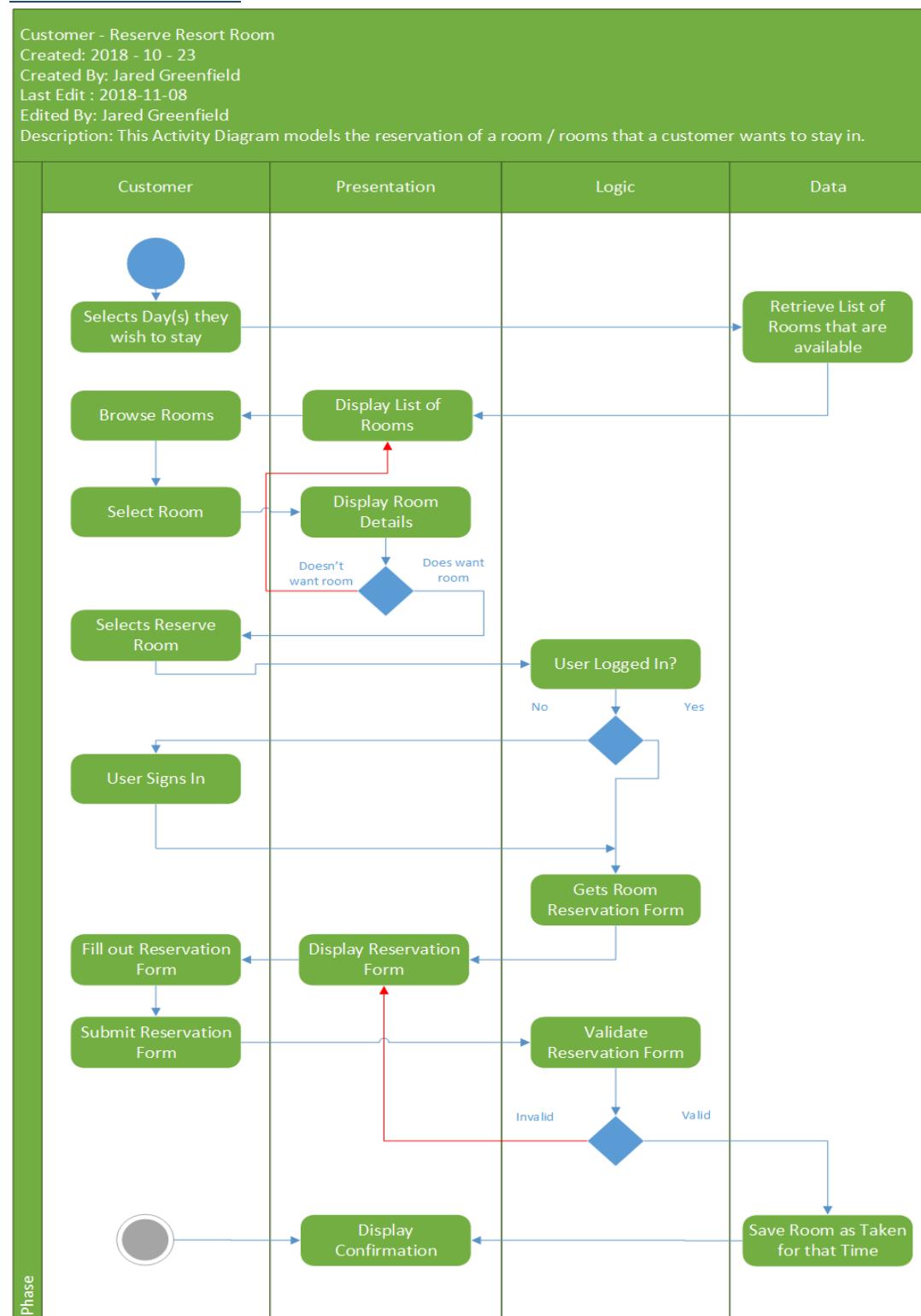
Created By: Jared Greenfield

Last Edit : 2018 - 10 - 26

Edited By: Jared Greenfield



## Reserve Resort Room



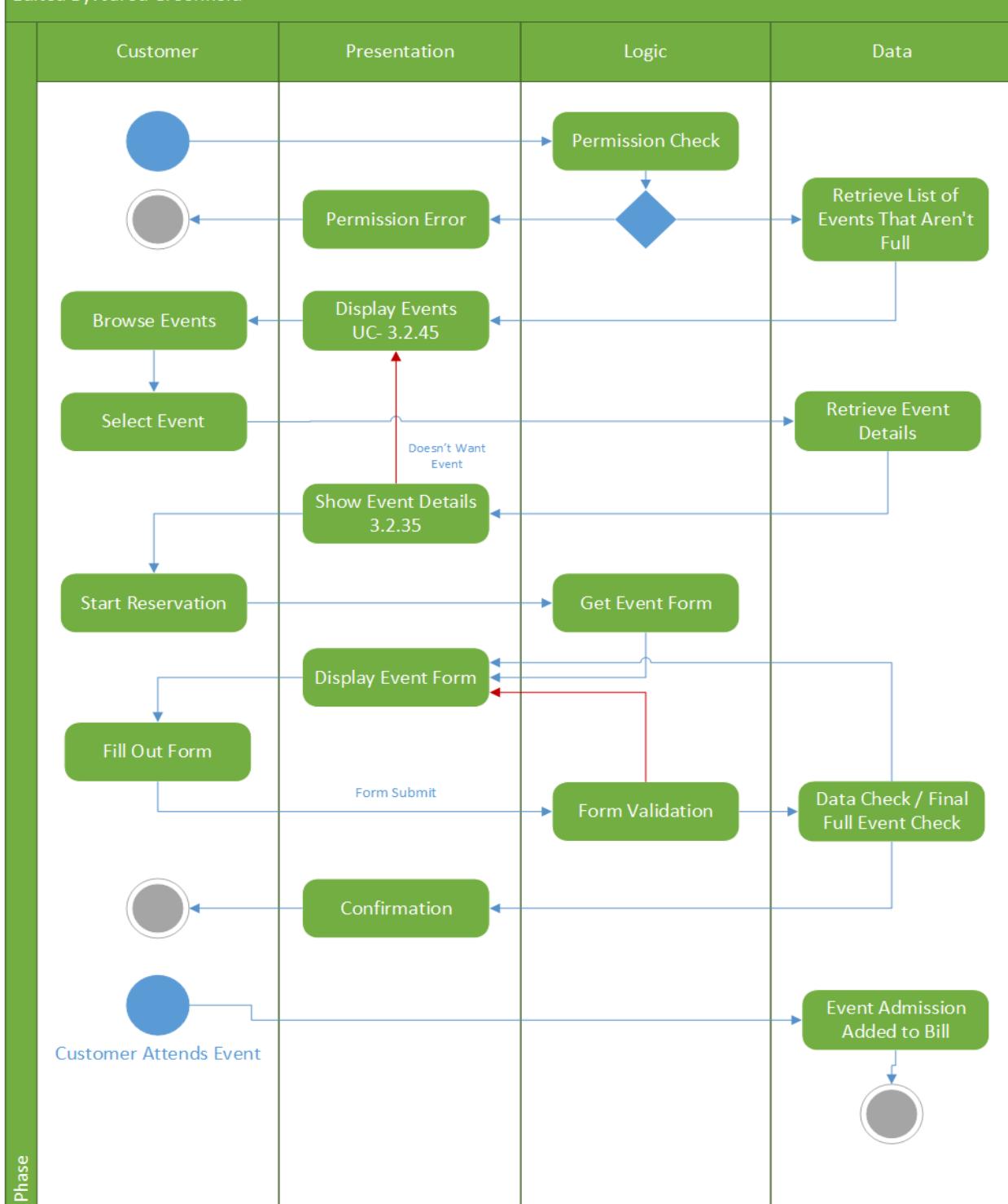
## Reserve Spot in Number Dependent Event

Created: 2018 - 10 - 17

Created By: Jared Greenfield

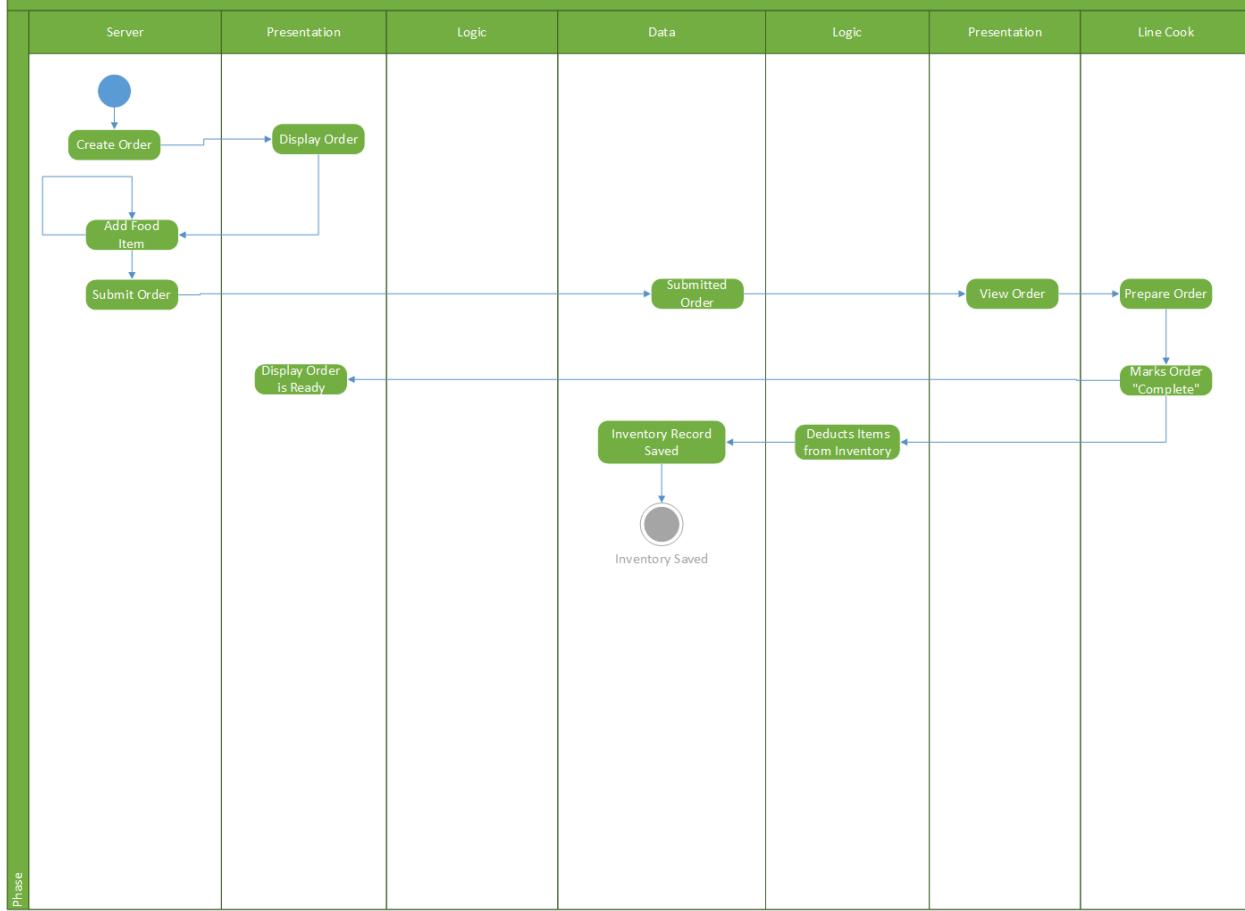
Last Edit : 2018 - 11 - 16

Edited By: Jared Greenfield



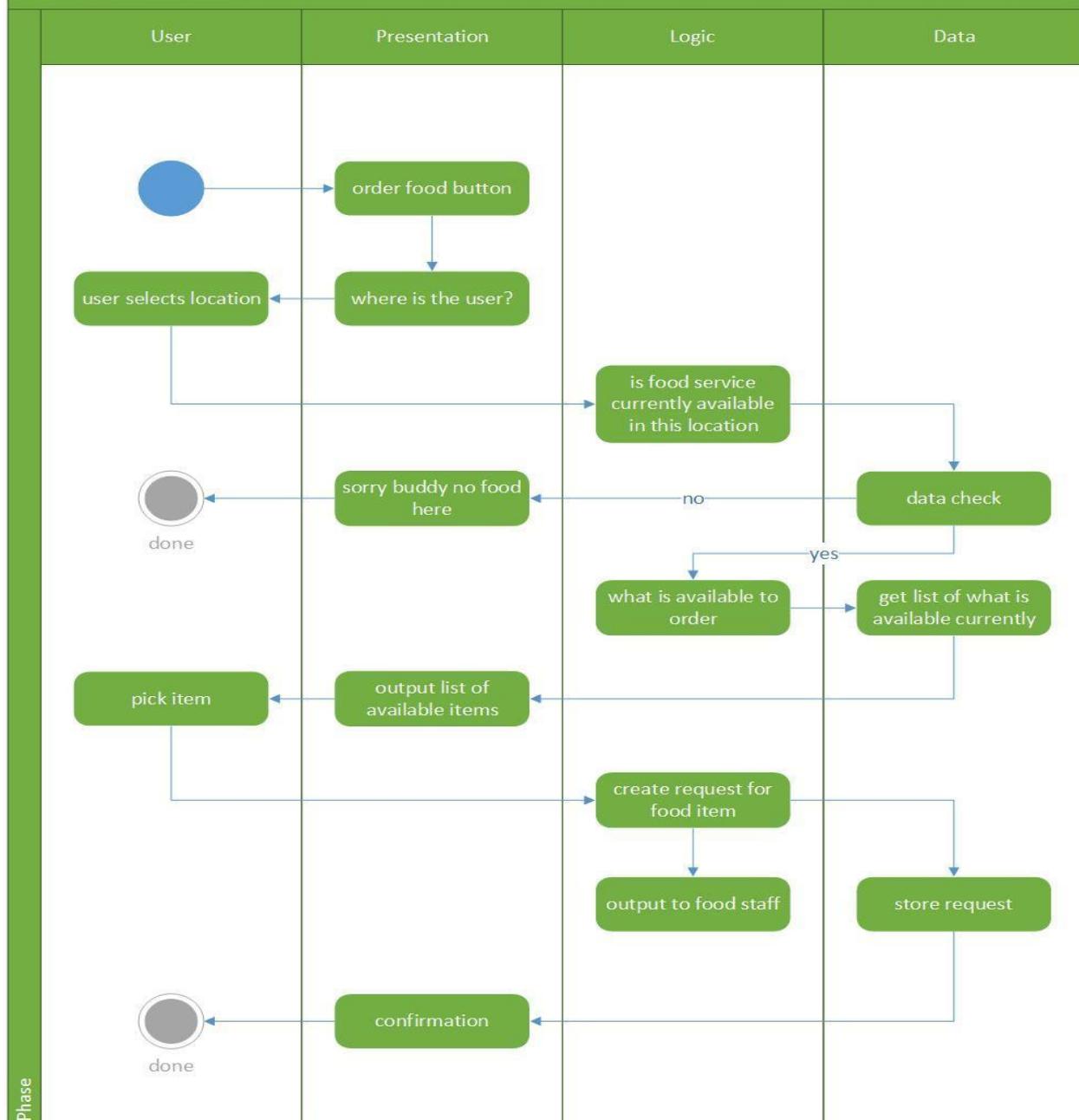
## Process Food Order

Process Food Order  
 Created By: Dani Russo  
 Date Created: 11/2/2018  
 Edited By:  
 Date Edited:



## Food Request

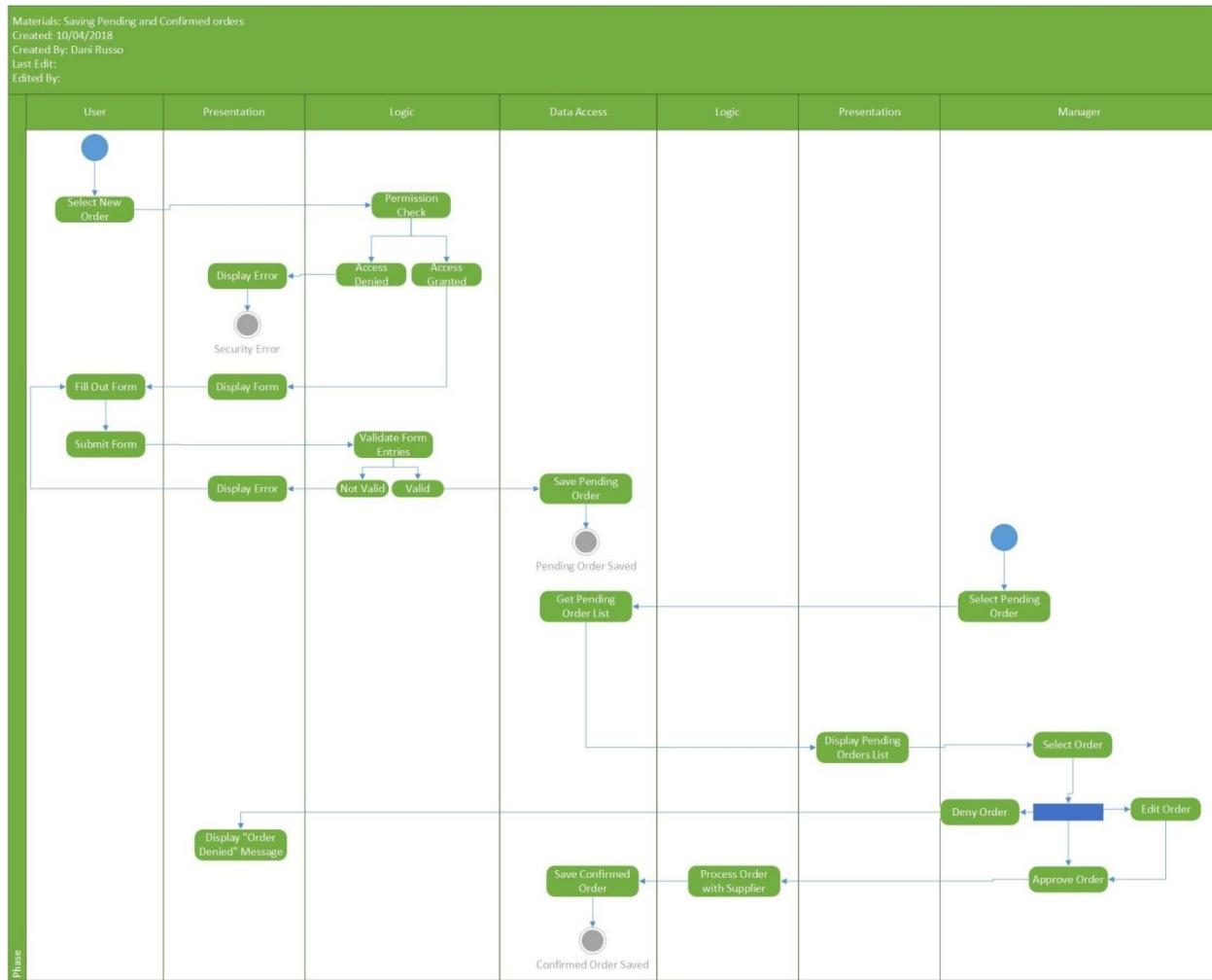
Order Food From Customer  
 Created: 2018 - 10 - 18  
 Created By: Austin Delaney  
 Last Edit: 2018 - 10 - 25  
 Edited by: Austin Delaney



## Materials

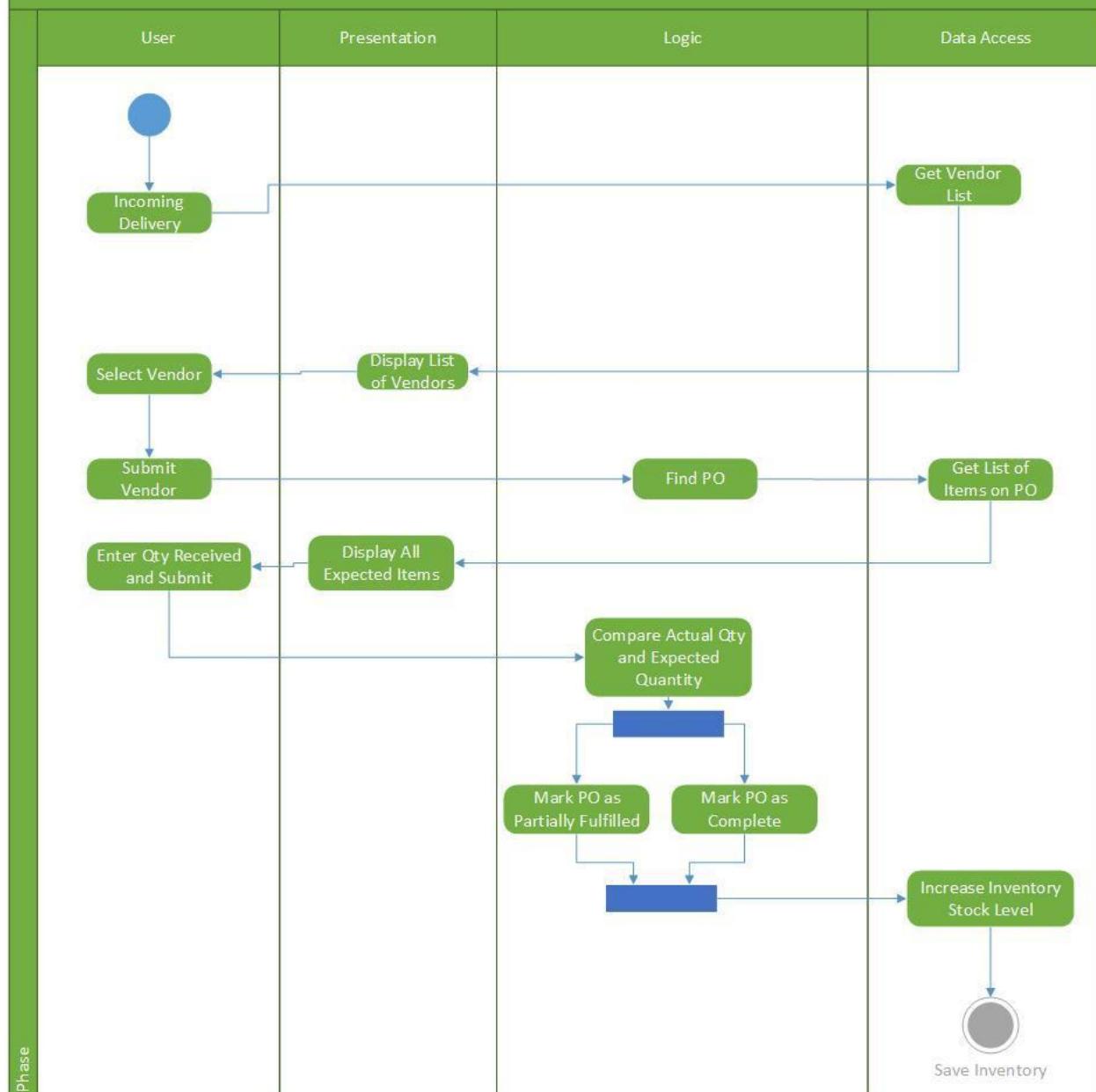
The following activity diagrams detail the progressive steps of managing the resort inventory, processing orders, creating and managing items, and vendor management. The diagrams highlight the information exchange between the user's initial action to the final step of updating the inventory.

### Confirm Order



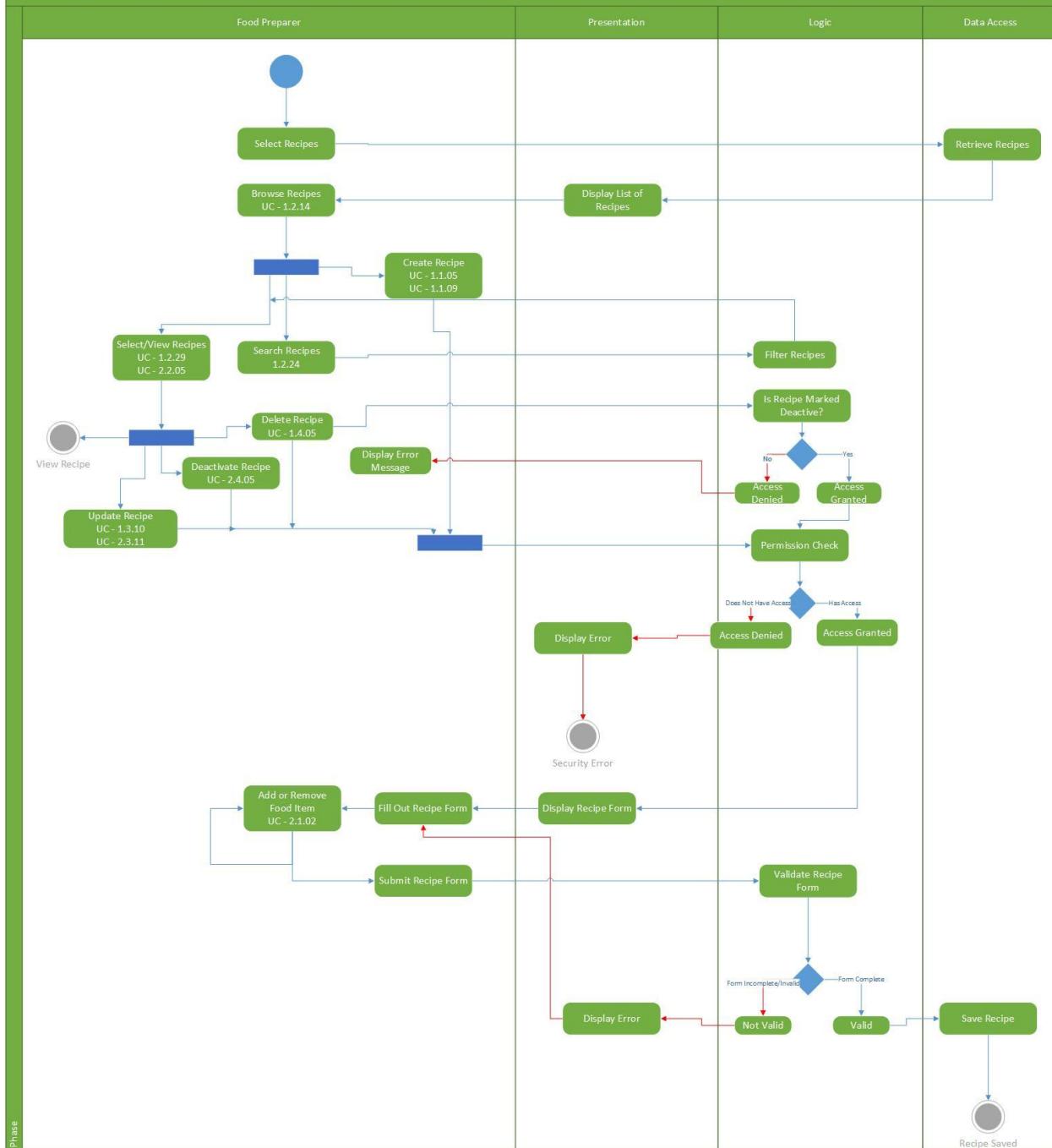
## Delivery

Materials: Processing Deliveries and Updating Inventory  
 Created: 10/04/2018  
 Created By: Dani Russo  
 Last Edit: 10/25/2018  
 Edited By: Dani Russo



## Recipes

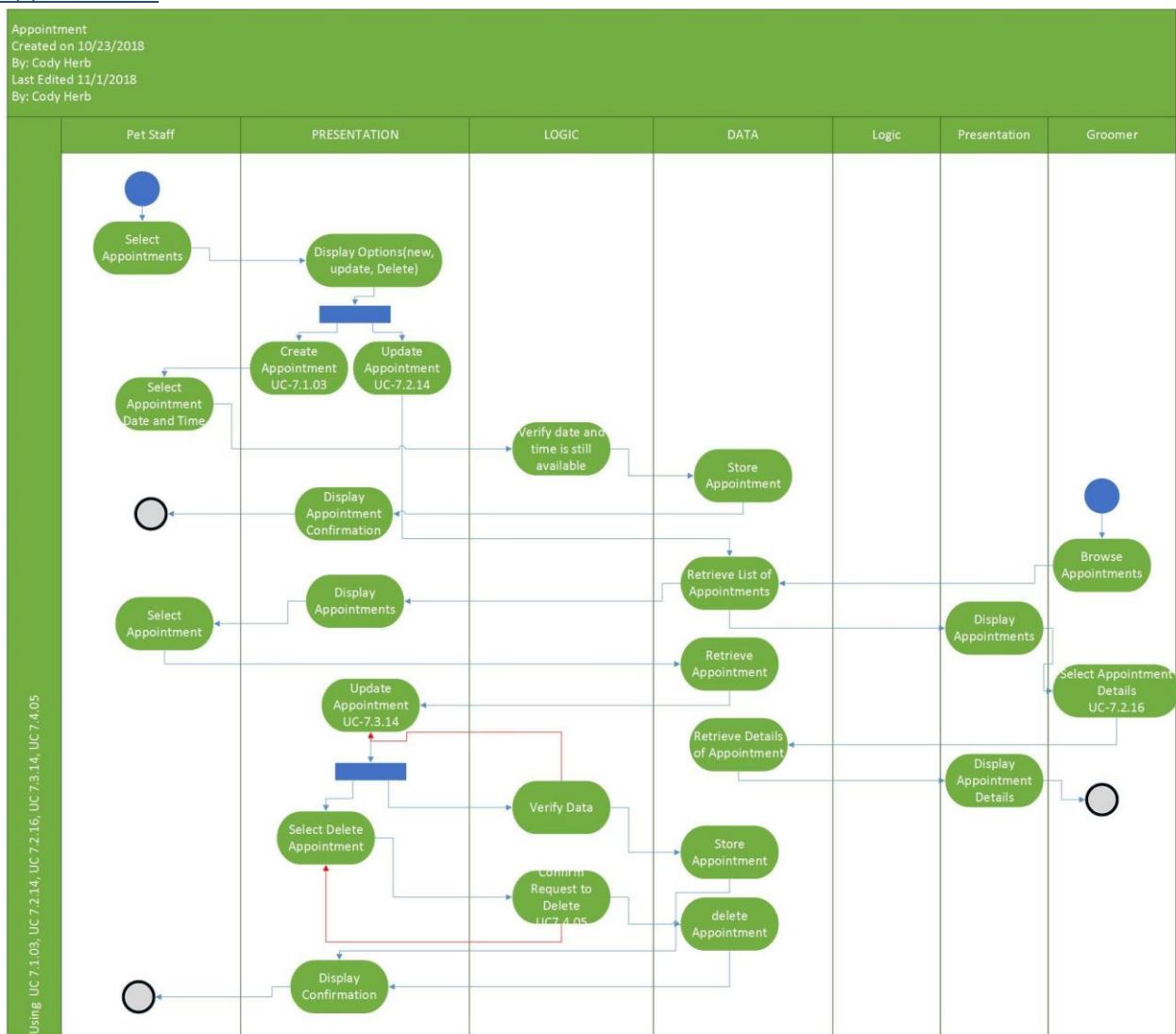
Materials: Saving New and Updated Recipes  
 Created: 10/22/2018  
 Created By: Dani Russo  
 Last Edit: 11/02/2018  
 Edited By: Dani Russo



## Personnel Management

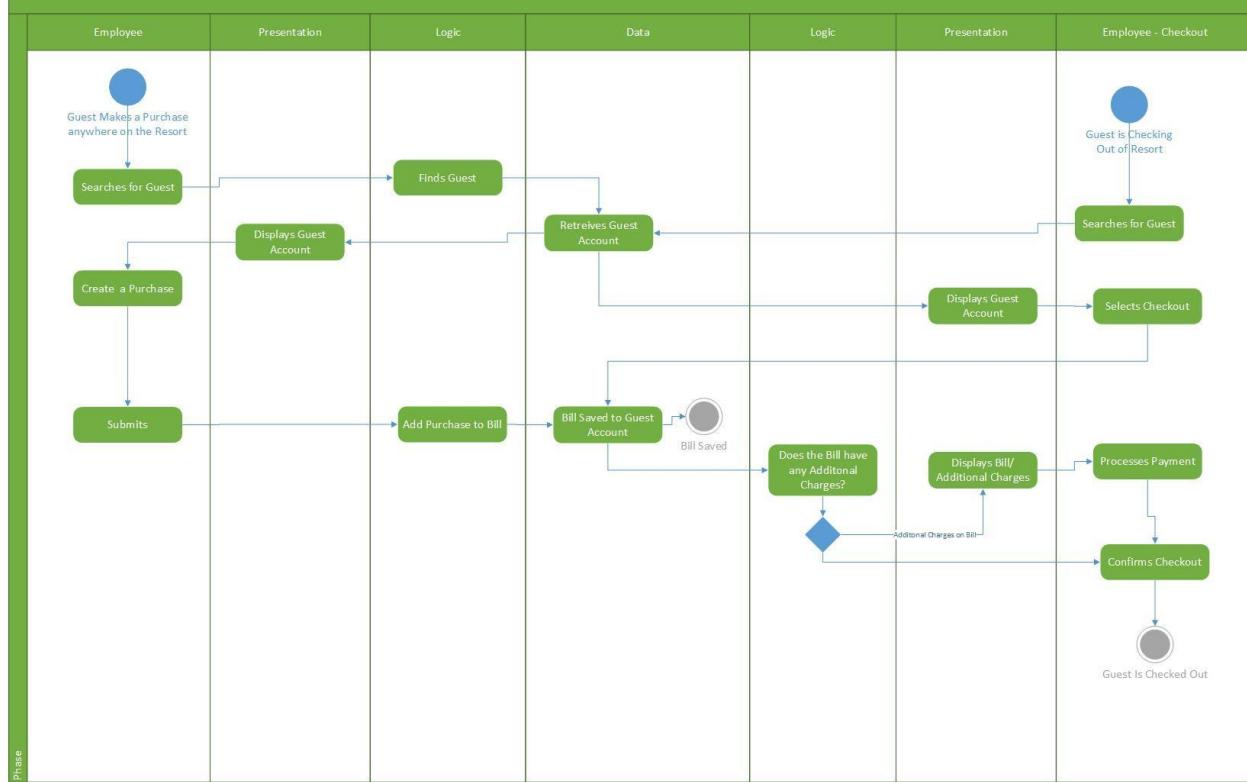
The following activity diagrams detail the flow of management and employee operations including customer refunds, supply orders, distributing incoming packages and employee scheduling. Any employee requests are saved as “pending” and are then transferred to management for approval.

## Appointment



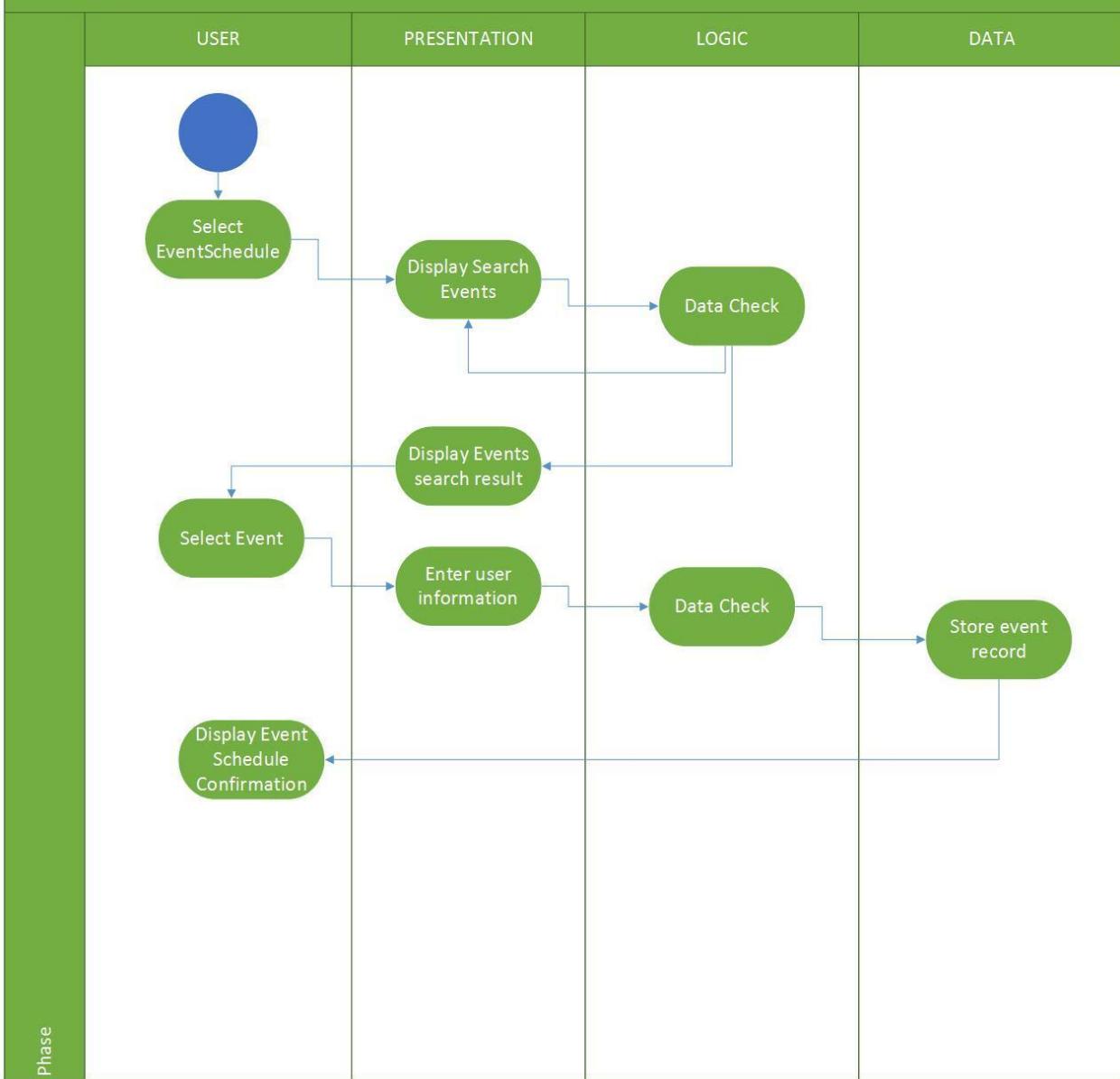
## Purchases and Checkout

Adding Items to a Bill and Checking Out  
 Created: 11/01/2018  
 Created By: Dani Russo  
 Last Edit:  
 Edited By:

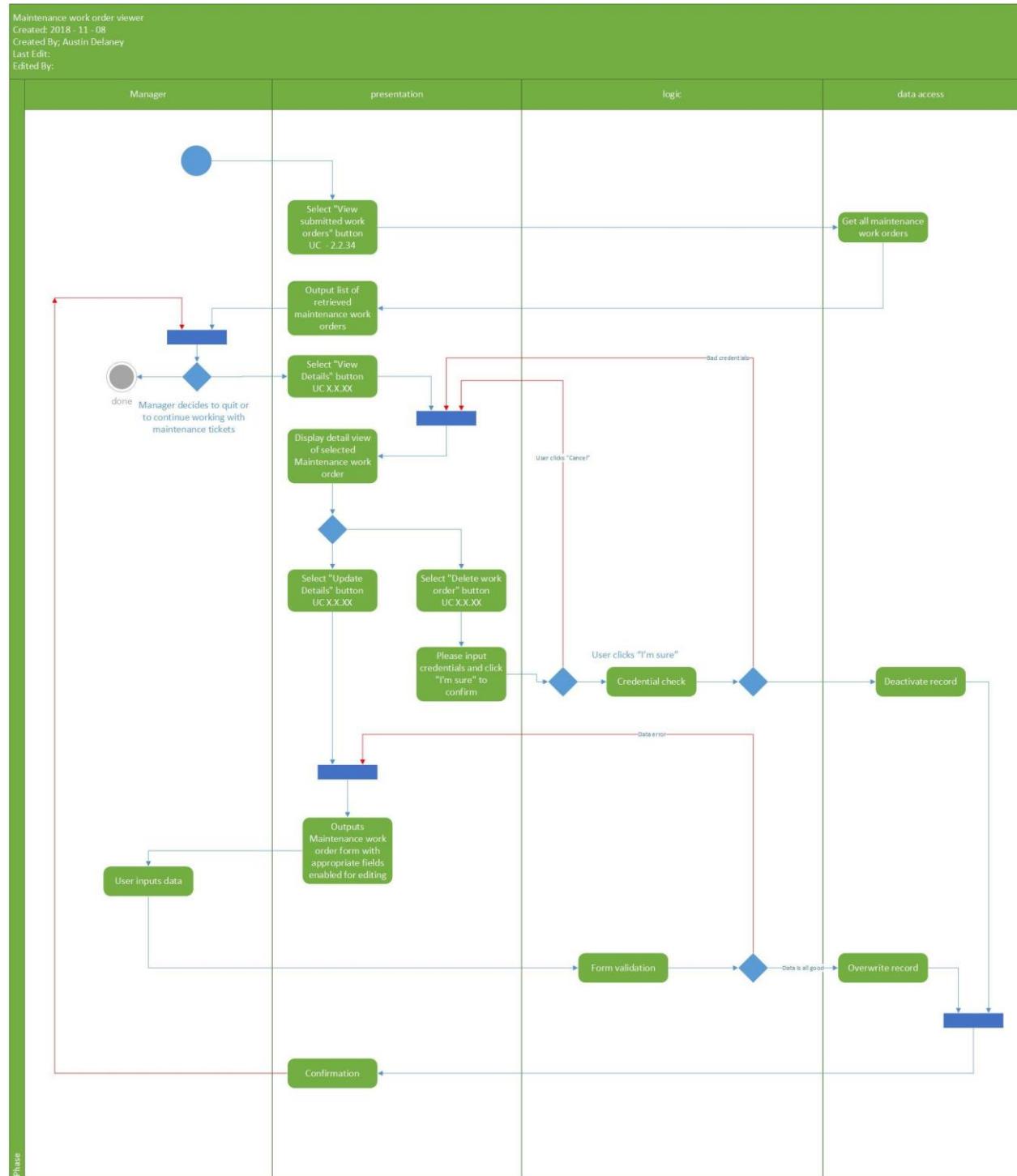


## Event Schedule

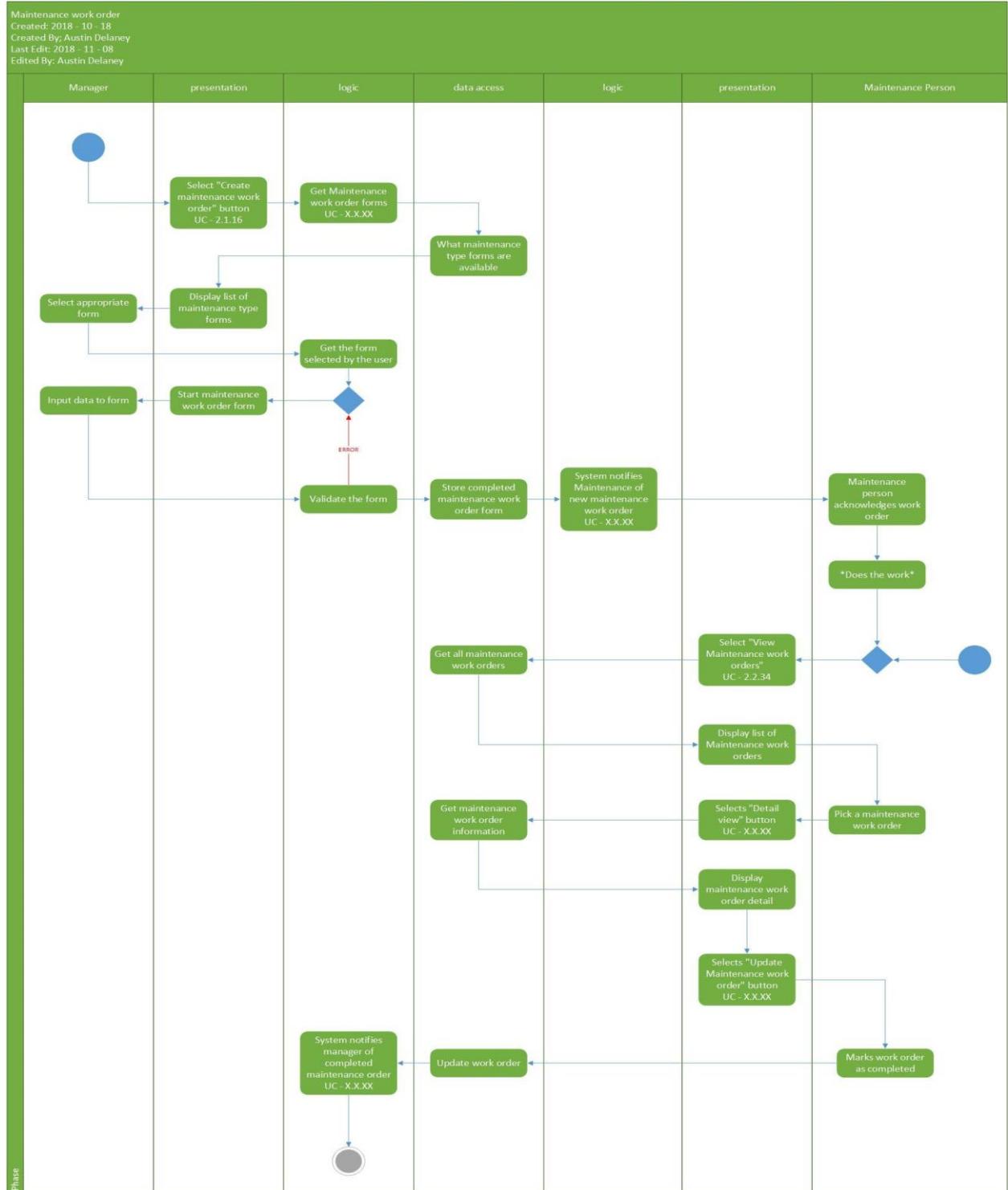
EventSchedule  
Last Edited 10/23/2018  
By: Cody Herb



## Maintenance Work Order Viewer

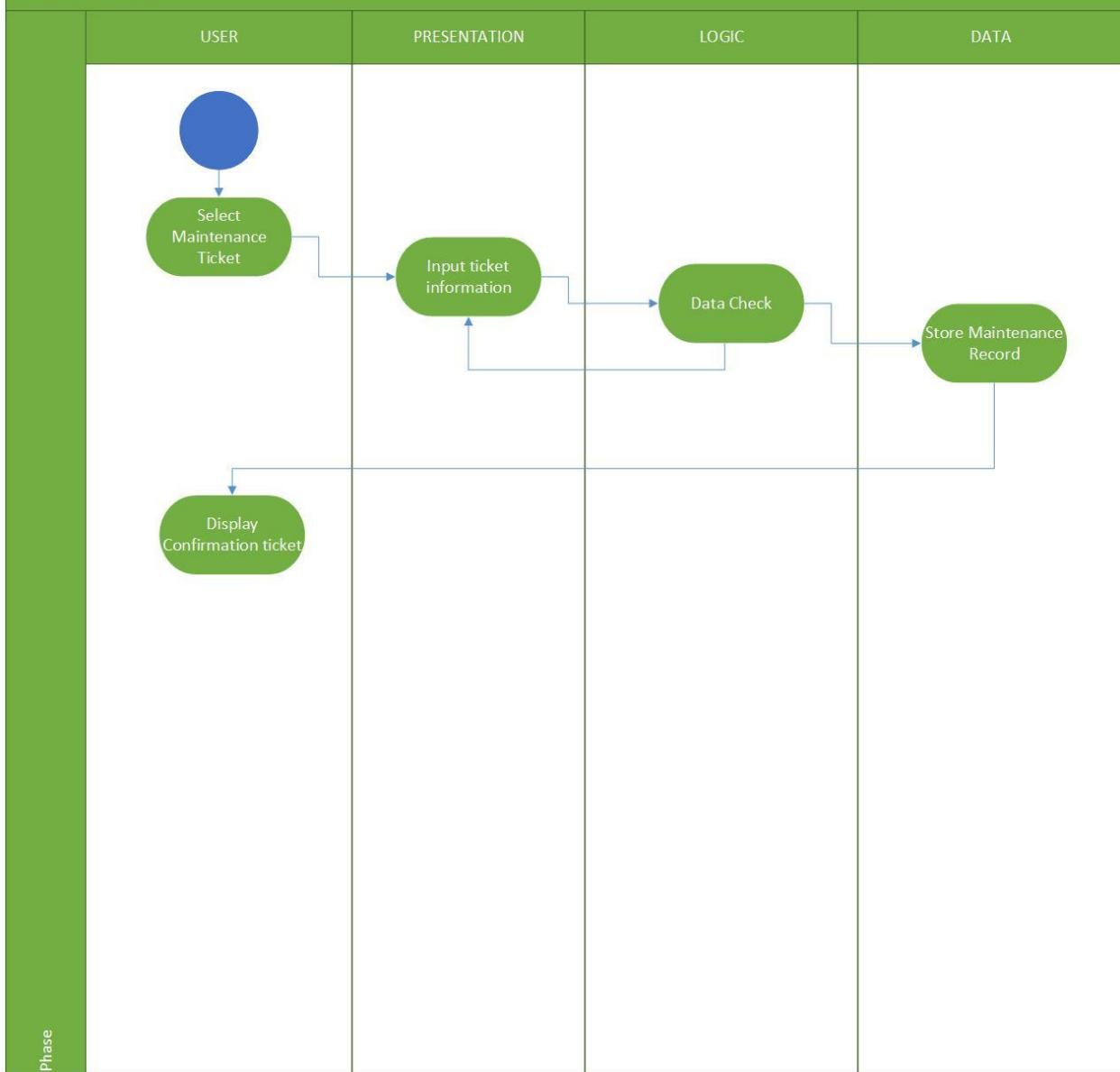


## Maintenance Work Order



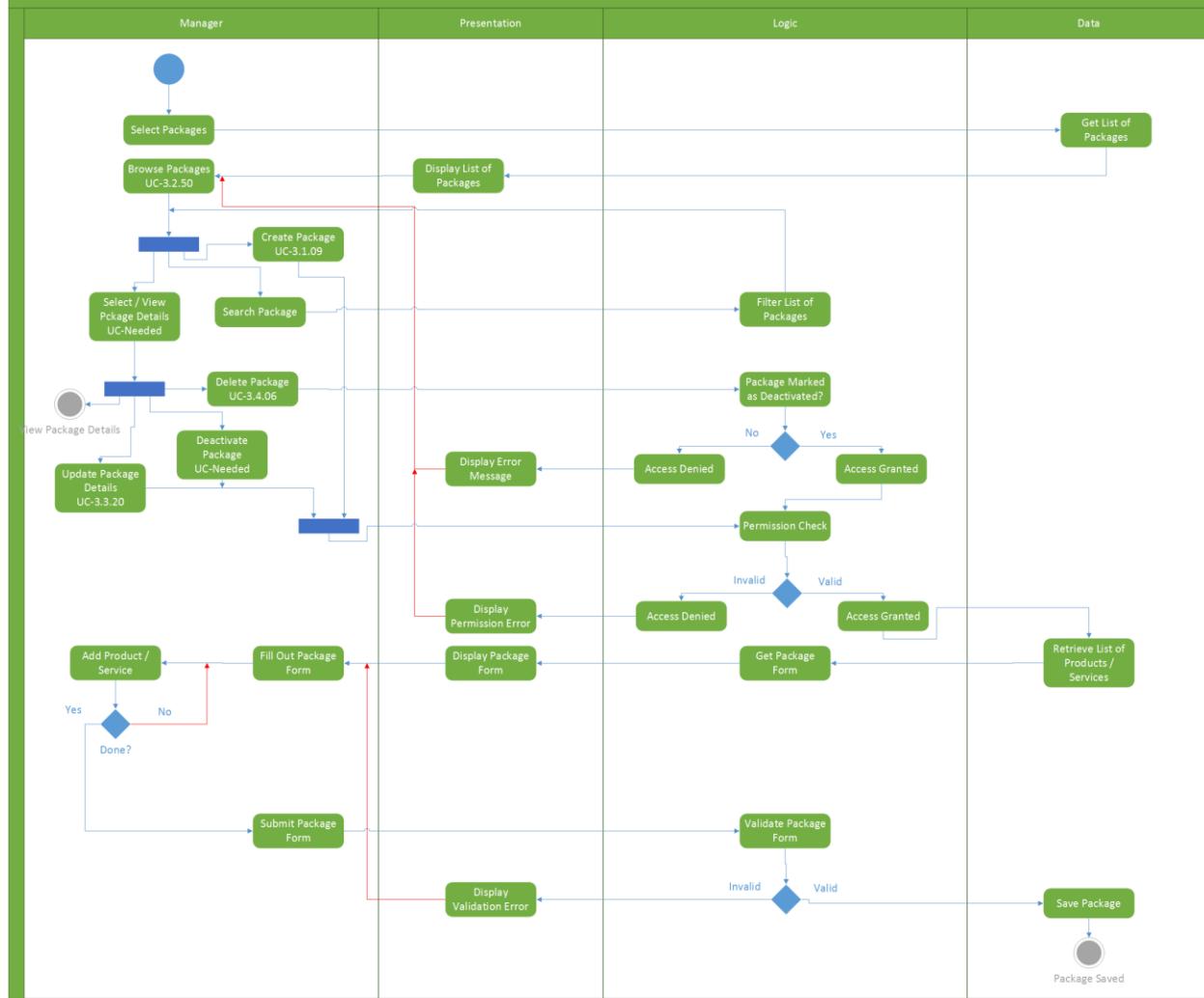
## Maintenance Ticket

MaintenanceTicket  
Last Edited 10/23/2018  
By: Cody Herb

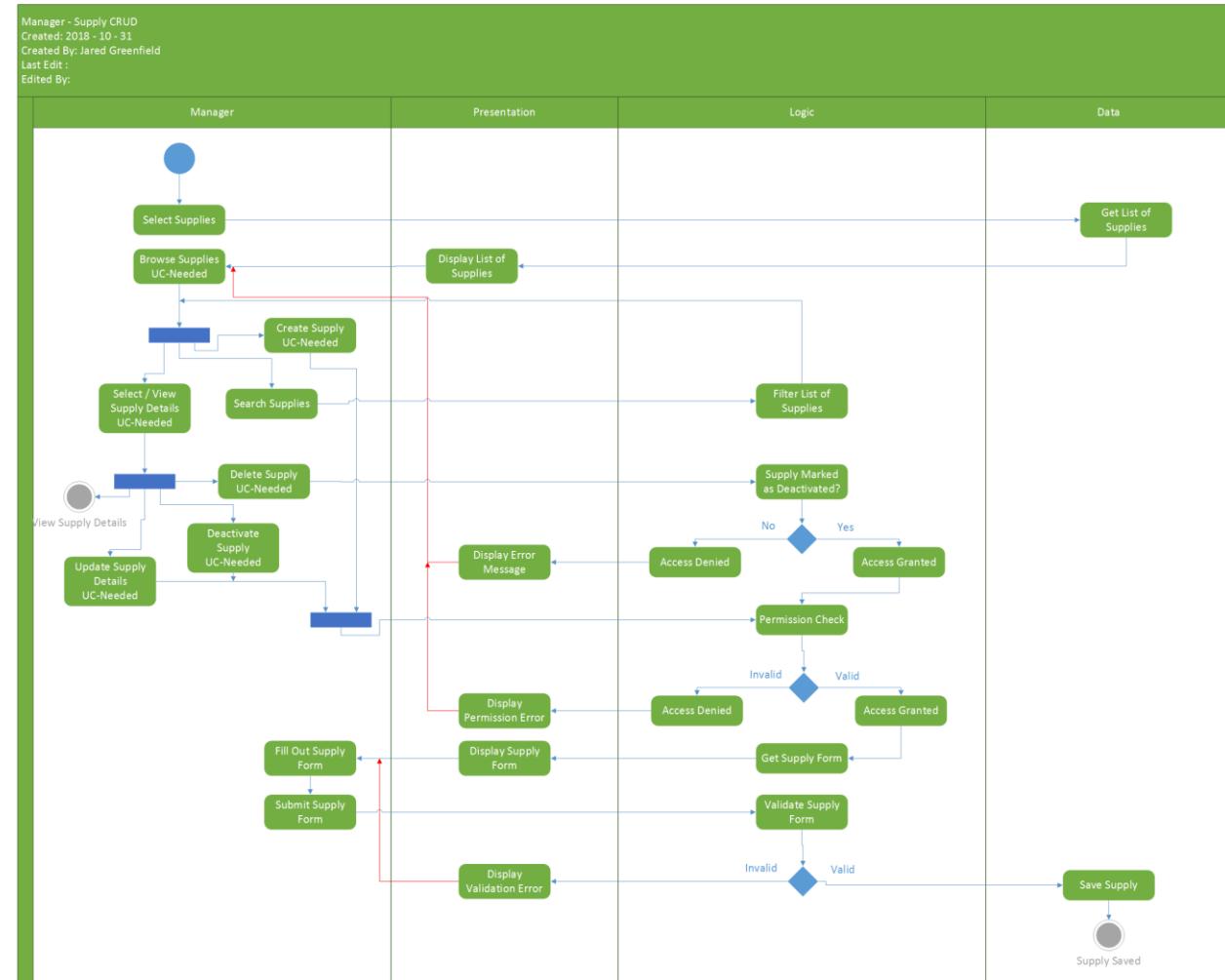


## Manager – Packages

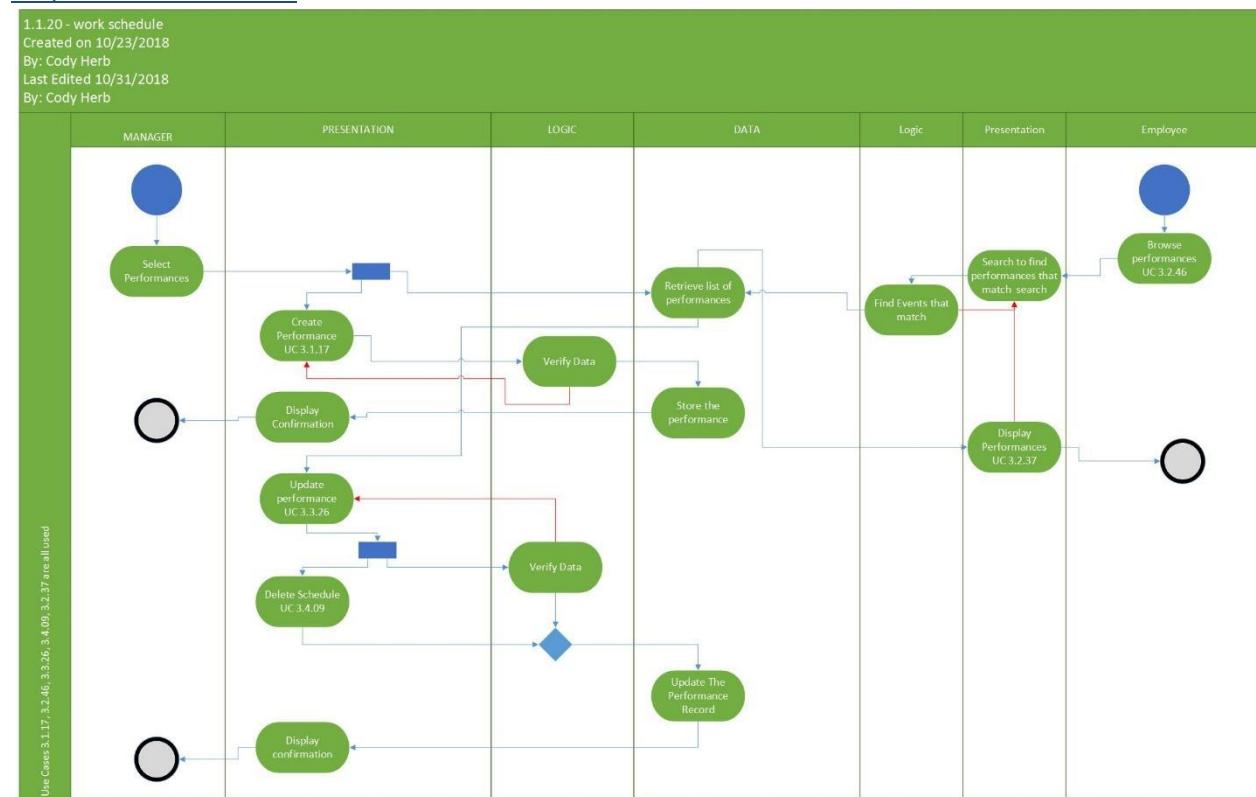
Manager - Package CRUD  
 Created: 2018-10-31  
 Created By: Jared Greenfield  
 Last Edit:  
 Edited By:



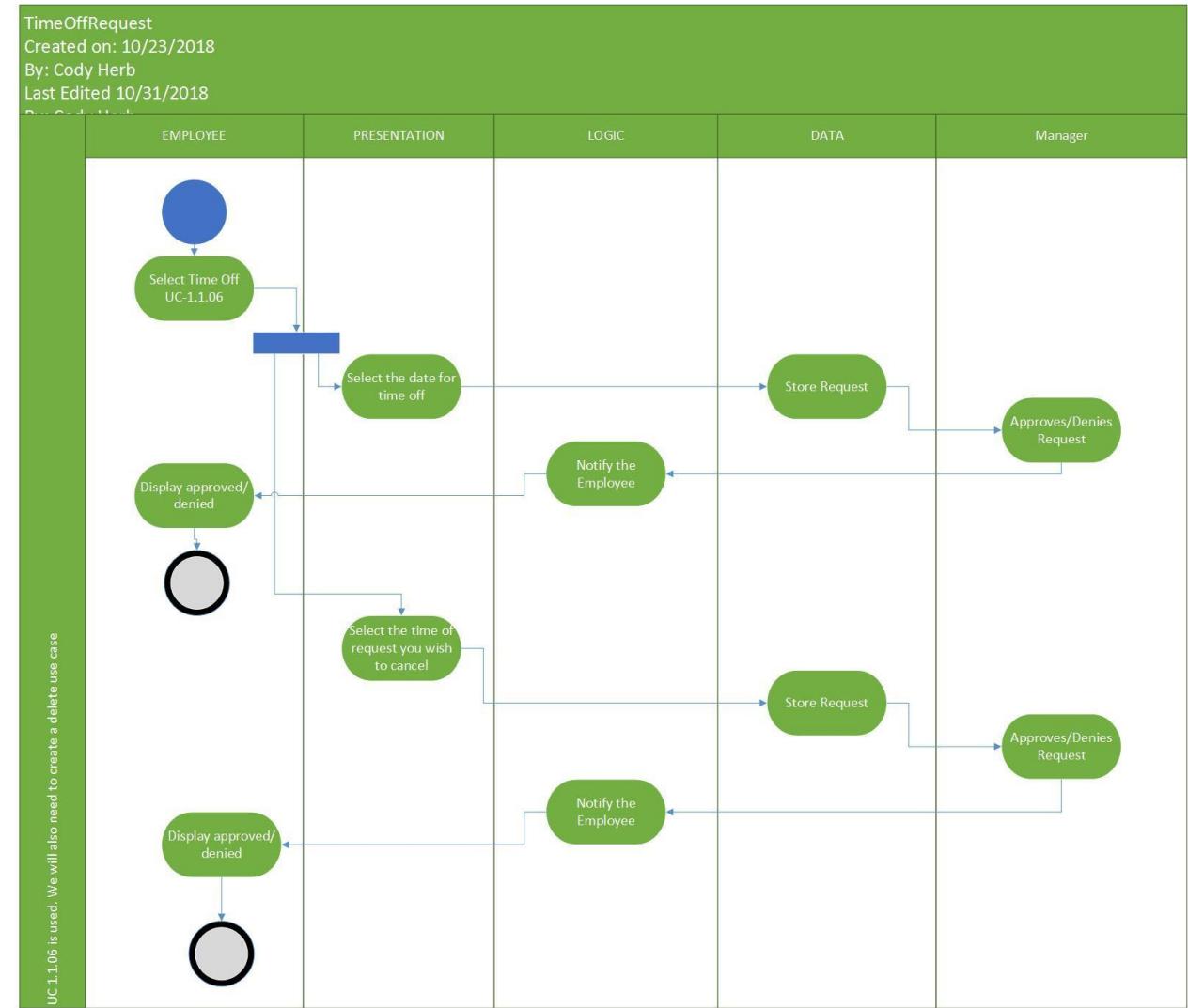
## Manager – Supplies



## Performance Schedule

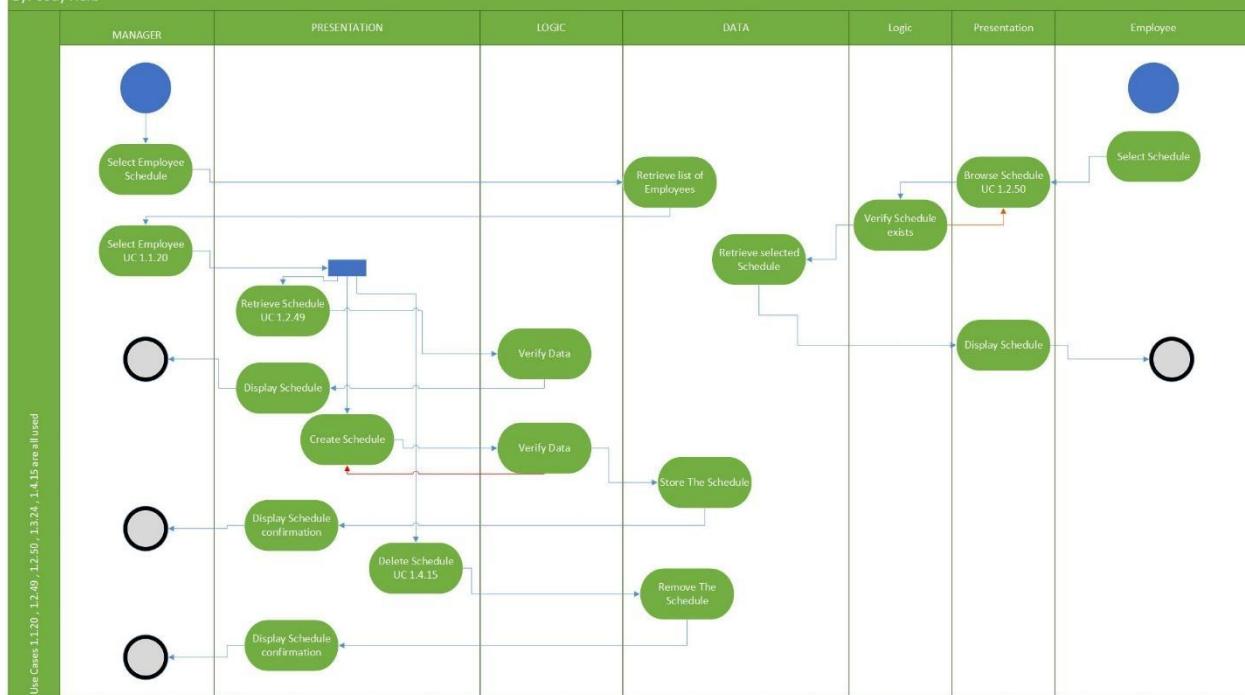


## Employee Time-Off Request



## Work Schedule

1.1.20 - work schedule  
 Created on 10/23/2018  
 By: Cody Herb  
 Last Edited 10/31/2018  
 By: Cody Herb



## Interfaces

### Internal

The internal interfaces are where internal users will interact with the individual functions of the solution. The design intent is consistency across all the interfaces to allow for a common look and feel and to make the user experience intuitive. This will make it easier to train new employees and to cross train existing employees.

#### Food Operations

New Ticket Order

Date:	<input type="text"/> / <input type="text"/>	Time:	<input type="text"/>	Waiter	<input type="text"/>	Table #	<input type="text"/> 3
Food Item	<input type="text"/>	Guest #:	<input type="text"/> 3	Price	<input type="text"/>		
Food Item	<input type="text"/>	Guest #:	<input type="text"/> 3	Price	<input type="text"/>		
New Item							
Tax	<input type="text"/>	Total Price		<input type="text"/>			
<input type="button" value="Cancel"/>				<input type="button" value="Save"/>			

**Review Full Tickets**

Search Filters:	<input checked="" type="checkbox"/> Incomplete <input type="checkbox"/> Complete	DateEntered: <input type="text" value=" / /"/> <input type="button" value="Calendar"/>	
Table #	Guest#	FoodItem	TimeOrdered

Select a food item to update instantly

Table #:	<input type="text" value="3"/> <input type="button" value="Up"/>	Food Item: <input type="text"/>	<input checked="" type="checkbox"/> Incomplete <input type="checkbox"/> Complete
Guest #:	<input type="text" value="3"/> <input type="button" value="Up"/>	<input type="button" value="Cancel"/> <input type="button" value="Add"/> <input type="button" value="Read/Update"/>	

If a food item is complete, it should automatically be deactivated or deleted, as it is not needed unless someone wishes to review all items that were made previously before.

**Update Ticket Order**

Date:	<input type="text" value=" / /"/> <input type="button" value="Calendar"/>	Time:	<input type="text"/>
Waiter	<input type="text"/>	Table #	<input type="text" value="3"/> <input type="button" value="Up"/>
Food Item	<input type="text"/>	Guest #:	<input type="text" value="3"/> <input type="button" value="Up"/>
Food Item	<input type="text"/>	Guest #:	<input type="text"/> <input type="button" value="Up"/>
<input checked="" type="checkbox"/> Incomplete <input type="checkbox"/> Complete			
Tax	<input type="text"/>	Total Price	<input type="text"/>

Read Ticket Order

Date:  /  

Time:

Waiter

Table #  3 

Food Item

Guest #:  3 

Price

Food Item

Guest #:  3 

Price

New Item 

Tax

Total Price

## Supply Operations

New Supply Order

Date:  /  

Product Number	<input type="text"/>	Description	<input type="text"/>	Qty	<input type="text"/>	Unit Price	<input type="text"/>
Product Number	<input type="text"/>	Description	<input type="text"/>	Qty	<input type="text"/>	Unit Price	<input type="text"/>

Add Item 

**Create Stock Record**

Stock ID	Stock Name	Description of Stock
Stock Type	Current Quantity 3 <input style="width: 20px; height: 20px;" type="button" value="▲"/>	Date Required <input style="width: 20px; height: 20px;" type="button" value="Calendar"/>
Employee Name	Employee Id	
Stock Supplier		

**Back** **Save**

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

**Browse Stock List**

<input type="checkbox"/> Active	<input type="checkbox"/> Inactive	<input type="radio"/> Filter 1	<input type="radio"/> Filter 2	<input type="radio"/> Filter 3	<input type="radio"/> Filter 4	search input
Stock ID	Stock Type	Stock Name	Qty	Employee Name	Supplier	Employee ID

**Cancel** **Delete** **Read/Update** **Add**

Read Supply Order

Product Number	<input type="text"/>	Description	<input type="text"/>	Qty	<input type="text"/>	Unit Price	<input type="text"/>
Product Number	<input type="text"/>	Description	<input type="text"/>	Qty	<input type="text"/>	Unit Price	<input type="text"/>
Add Item 							

Read Stock Record

Stock ID	<input type="text"/>	Stock Name	<input type="text"/>	Description of Stock	
Stock Type	<input type="text"/>	Current Quantity	<input type="text" value="3"/> 		
Employee Name	<input type="text"/>	Employee Id	<input type="text"/>	Date Required	<input type="text"/> 
Stock Supplier	<input type="text"/>				

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

<input type="checkbox"/> Active	<input type="checkbox"/> Inactive	<input type="radio"/> Completed	<input type="radio"/> In Progress	<input type="radio"/> Out of stock	<input type="radio"/> Filter 4	search input
Req ID	Date	Equip ID	Vendor ID	Description		

**Supply Order**

Product Number	<input type="text"/>	Description	<input type="text"/>	Qty	<input type="text"/>	Unit Price	<input type="text"/>
Product Number	<input type="text"/>	Description	<input type="text"/>	Qty	<input type="text"/>	Unit Price	<input type="text"/>
<input style="font-size: small; border: 1px solid black; border-radius: 50%; padding: 2px 10px; margin-right: 5px;" type="button" value="Add Item"/>							
<input type="radio"/> Activate <input type="radio"/> Deactivate							
<input style="border: 1px solid black; padding: 2px 10px;" type="button" value="Back"/>				<input style="border: 1px solid black; padding: 2px 10px;" type="button" value="Save"/> <input style="border: 1px solid black; padding: 2px 10px;" type="button" value="Edit"/>			

**Update Stock Record**

Stock ID	12345678	Date Required	<input type="text"/>		Description of Stock
Stock Type	Some text	Stock Name	<input type="text"/>		
Current Quantity <input type="text" value="3"/>					
Employee Id <input type="text" value="Some text"/>					
Employee Name <input type="text" value="Some text"/>					
<input type="radio"/> Activate <input type="radio"/> Deactivate					
<input style="border: 1px solid black; padding: 2px 10px;" type="button" value="Back"/> <input style="border: 1px solid black; padding: 2px 10px;" type="button" value="Save"/> <input style="border: 1px solid black; padding: 2px 10px;" type="button" value="Edit"/>					

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

Supplier Information Create

Supplier Name	<input type="text"/>	Supplier Type	<input type="button" value="▼"/>
Street Address	<input type="text"/>		
Street Address 2	<input type="text"/>		
City	<input type="text"/>		
State	<input type="text"/>		
Zip	<input type="text"/>		
Date Active	<input type="text"/> / <input type="text"/> / <input type="button" value="Calendar"/>		
E-Mail Contact:	<input type="text"/>		
Phone # Contact:	<input type="text"/>		
Supplier Notes			
<input type="button" value="Cancel"/>		<input type="button" value="Save"/>	

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

Browse List of Suppliers

<input type="radio"/> Active	SupplierName: <input type="text"/>	SupplierCity: <input type="text"/>	Type: <input type="button" value="▼"/>	<input type="button" value="Search"/>
<input type="radio"/> Inactive				
List of Suppliers				
SupplierName	SupplierType	SupplierCity	PhoneNumber	Notes
<input type="button" value="Cancel"/>		<input type="button" value="Add"/> <input type="button" value="Read/Update"/> <input type="button" value="Delete"/>		

Read Supplier

Supplier Name

Street Address

Street Address 2

City

State

Zip

Supplier Type

Supplier Notes

Date Active  /

E-Mail Contact:

Phone # Contact:

[Back](#)

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

Update Supplier Information

Supplier Name

Street Address

Street Address 2

City

State

Zip

Supplier Type

Supplier Notes

Date Active  /

E-Mail Contact:

Phone # Contact:

Activate  Deactivate

[Back](#)

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

**Deactivation/Delete Confirmation**

SupplierName:	<input type="text"/>
Type:	<input type="text"/> 
City:	<input type="text"/> <input type="text"/> / 

Are you sure you would like to  
deactivate/delete this supplier?

Facility Operations

**ReadCart**

<b>Cart</b>	
Price	
Product A	\$\$\$\$
<a href="#">Delete</a> <a href="#">Details</a>	
Product A	\$\$\$\$
<a href="#">Delete</a> <a href="#">Details</a>	
Subtotal	\$\$\$\$
<input type="button" value="Checkout"/>	

### New Maintenance Ticket

Date Requested  

Requester

Problem Type

Location

Priority

Details

**New Package**

Package ID	<input type="text"/>	Package Start <input type="text"/> / <input type="button" value="Calendar"/>
Package Name	<input type="text"/>	Package end <input type="text"/> / <input type="button" value="Calendar"/>
Package Type	<input type="text"/>	Package Description <input type="text"/>
Package Quantity	<input type="text" value="3"/> <input type="button" value="Up"/> <input type="button" value="Down"/>	
Total Cost	<input type="text"/>	

### New Building

Building ID:

Description

Building Name:

Building Type:

Location:

Date Build:  /  

Purpose:

### Read Building Details

Building ID:

Description

Building Name:

Building Type:

Location:

Date Build:  /  

Purpose:

Browse Packages

Active  Inactive

List of Orders

search

Id	Package Name	PackageType	Date Start	Date End	Pack Quantity	Total Cost

Cancel  Add  Read/Update  Delete

### Read Package

Package Start  / / 

Package ID

Package end  / / 

Package Name

Package Description

Package Type

Package Quantity  3 

Total Cost

Browse Buildings

Active  Inactive  Filter 1  Filter 2  Filter 3  Filter 4

search

Bldg ID	Bldg Type	Location	Date Built	Misc	Description
---------	-----------	----------	------------	------	-------------

Double click on the data grid to bring up the record

Add New      Edit Detail      Exit

**Update Package**

Package ID	<input type="text"/>	Package Start	<input type="text"/> / / 
Package Name	<input type="text"/>	Package end	<input type="text"/> / / 
Package Type	<input type="text"/>	Package Description	
Package Quantity	3 	<input type="text"/>	
Total Cost	<input type="text"/>	<input checked="" type="checkbox"/> Active <input type="checkbox"/> Deactive	
<b>Back</b>		<b>Edit</b>	<b>Save</b>

Update Building

Building ID:	<input type="text"/>	Description
Building Name:	<input type="text"/>	
Building Type:	<input type="text"/>	
Location:	<input type="text"/>	
Date Build:	<input type="text"/> / <input type="text"/> 	
Purpose:	<input type="text"/>	

**Cancel**

**Save**

**Deactivation/Delete Confirmation**

PackageID:	<input type="text"/>	Name:	<input type="text"/>
Type:	<input type="text"/> 	StartDate:	<input type="text"/> / <input type="text"/> 

Are you sure you would like to  
deactivate/delete this package?

**Cancel**

**Deactivate/Delete**

### Deactivation Confirmation

Bldg ID:  Name:   
Location:  Type:

Are you sure you would like to deactivate this building?

### Create Product

Product Name:	<input type="text"/>	Product Description
Cost to make:	<input type="text"/> \$0.00	
Selling price:	<input type="text"/> \$0.00	
Amount sold:	<input type="text" value="0"/> <input type="button" value="▼"/>	
Amount to keep stocked:	<input type="text" value="0"/> <input type="button" value="▼"/>	
Amount Stocked:	<input type="text" value="0"/> <input type="button" value="▼"/>	

### Read Product

Product Name: Hot Sauce

Product Description

Tasty Stuff

Cost to make: \$0.50

Selling price: \$2.99

Amount sold: 507

Amount to keep stocked: 50 ▲▼

Amount Stocked: 50 ▲▼

Back

**Browse Products**

List of Products				
<input type="checkbox"/> Active	<input type="checkbox"/> Deactive			
ProductName	ProductCost	AmountSold	InStock	Description

search

Cancel   Add   Read/Update   Delete

Update Product

Product Name:	<input type="text"/>
Cost to make:	<input type="text" value="\$0.0"/>
Selling price:	<input type="text" value="\$0.0"/>
Amount sold:	<input type="text" value="0"/> 
Minimum Stock Allowed:	<input type="text" value="3"/> 
Amount Stocked:	<input type="text" value="0"/> 

[Back](#) [Edit](#) [Save](#)

Deactivation Confirmation

ProductID:	<input type="text"/>
Name:	<input type="text"/>
SellPrice:	<input type="text"/>
InStock:	<input type="text"/>

Are you sure you would like to deactivate this product?

[Cancel](#) [Deactivate](#)

**View Maintenance Tickets**

Filter:  Not Fixed  Fixed  Both

Requestor	Priority	DateRequested	ProblemType	Location	Fixed?	MaintenanceDetails

Double click on the data grid to bring up the record

**[View Detail](#)**   **[Deactivate](#)**   **[Exit](#)**

**Receptionist - View/Update Maintenance Ticket**

<p>Requestor <input type="text"/></p> <p>Assigned to <input type="text"/></p> <p>Date submitted <input type="text"/> </p> <p>Problem Type <input type="text"/></p> <p>Location <input type="text"/> </p> <p>Priority</p> <ul style="list-style-type: none"> <li><input type="radio"/> Immediate</li> <li><input type="radio"/> High</li> <li><input type="radio"/> Medium</li> <li><input type="radio"/> Low</li> </ul> <p>Fixed?</p> <ul style="list-style-type: none"> <li><input type="radio"/> No</li> <li><input type="radio"/> Yes</li> </ul>	<p><b>Maintenance Details</b></p> <p>Description of request</p> <div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>
---	---

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This will also come up when you create a new record, only the contents will be blank.)

**[Edit](#)**   **[Save](#)**   **[Back to List](#)**

**Deactivation Confirmation**

SlipID:	<input type="text"/>	DateSubmitted:	<input type="text"/> / / <input type="button" value="Calendar"/>	
Requestor:	<input type="text"/>	AssignedTo:	<input type="text"/>	

Are you sure you would like to  
deactivate this maintenance  
slip?

<input type="button" value="Cancel"/>	<input type="button" value="Deactivate"/>
---------------------------------------	---

### Guest Operations

**New Room**

<p>Room Number <input type="text"/></p> <p>Building <input type="text"/></p> <p>Type <input type="text"/></p> <p>Max Capacity: <input style="width: 20px; height: 20px; border: 1px solid black; text-align: center; padding: 0 5px;" type="text" value="3"/></p> <p>Occupied? <input checked="" type="radio"/> No <input type="radio"/> Yes</p> <p>Needs Cleaned? <input checked="" type="radio"/> No <input type="radio"/> Yes</p>	<p>Description of Room       </p> <p>Guest Info</p> <p>Last Name <input type="text"/></p> <p>First Name <input type="text"/></p> <p>Check-In <input type="text"/> / / <input type="button" value="Calendar"/></p> <p>Check-Out <input type="text"/> / / <input type="button" value="Calendar"/></p>
--	---

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

### New Account

Employee ID 12345678

First Name:

Last Name:

Role:

Password:

Re-Type Password:

Date Of Birth:  

E-Mail:

Phone Number:

Profile Picture



 Upload Picture

Emergency Contact

First Name:

Last Name:

Phone Number:

### Read Account

Employee ID 12345678

First Name:

Last Name:

Role:

Password:

Re-Type Password:

Date Of Birth:  

E-Mail:

Phone Number:

#### Profile Picture



 Upload Picture

#### Emergency Contact

First Name:

Last Name:

Phone Number:

Browse Accounts

( search)

Active    Inactive    Filter 1    Filter 2    Filter 3    Filter 4

Last Name	First Name	Role	DOB	Date Created

**Update Account**

Employee ID 12345678

First Name:

Last Name:

Role:

Password:

Re-Type Password:

Date Of Birth:

E-Mail:

Phone Number:

Profile Picture

**Emergency Contact**

First Name:

Last Name:

Phone Number:

**Deactivation Confirmation**

AccountID:  UserName:

DateJoined:  /  AccountType:

Are you sure you would like to deactivate this account?

### New Guest Vehicle

Ticket #

Car Description:

License Plate

*It is important to list any damages when vehicle is received to avoid fraud claims when given possession.*

Make

Model

Color

#### Dropped Off

Date Arrived  / /

Date Departed  / /

Time Arrived

Time Departed

Employee

Employee

Location

Parking #  3

### Create Shuttle

Shuttle VIN:	<input type="text"/>	Shuttle Description:	<input type="text"/>
License Plate:	<input type="text"/>		
Shuttle Type:	<input type="text"/>		
Shuttle Color:	<input type="text"/>		
Shuttle Manufacturer:	<input type="text"/>		
Shuttle Name:	<input type="text"/>	<input type="checkbox"/> In Use <input type="checkbox"/> In Maintenance	
Seats:	4 <input type="button" value="▼"/>		
Mileage:	<input type="text"/>		
Driver Name:	<input type="text"/>		
Destination:	<input type="text"/>		
Date Arrived:	<input type="text"/> / / <input type="button" value="Calendar"/>		

Retrieve Shuttle Info

**Filters:**

Owner Name: <input type="text"/>	License Plate: <input type="text"/>	Location: <input type="text"/>
Employee Name: <input type="text"/>	Vehicle Name: <input type="text"/>	Vehicle Type: <input type="text"/>

---

**List of Vehicles**

License #	Owner:	Vehicle Name:	Vehicle Type:	Location:	Parking #:	Description

The User may specifically want to search a field, ex. Guest name, and view the details of what car they have in their name to bring the correct vehicle to the correct guest.

### Read Shuttle

Shuttle VIN	01234567890123456	Shuttle Description:
License Plate	123 XYZ	This is a very opt description of this vehicle.
Shuttle Type	Autobus	
Shuttle Color	Yellow	
Shuttle Manufacturer	General Atomics	
Shuttle Name:	Big Battle Bus	<input checked="" type="checkbox"/> In Use <input type="checkbox"/> In Maintenance
Seats:	80	
Mileage	45600	
Driver Name	Tim Dalton	
Destination	Candy Shop	
Date Arrived	01/01/199	<input checked="" type="checkbox"/> Active
<input type="button" value="Back"/>		<input type="button" value="Edit"/> <input type="button" value="Save"/>

**Browse Rooms**

<input type="checkbox"/> Active	<input type="checkbox"/> Inactive	Location <input type="button" value="▼"/>	Type <input type="button" value="▼"/>	<input type="checkbox"/> Capacity	<input type="checkbox"/> Occupied	<input type="text" value="search input"/>
<input type="checkbox"/> Needs Cleaned						
Room#	Location	Type	Max Size	Occupied	Needs Cleaned	Description

<input type="button" value="Exit"/>	<input type="button" value="Delete"/>	<input type="button" value="Read/Update"/>	<input type="button" value="Add"/>
-------------------------------------	---------------------------------------	--	------------------------------------

**Read Room**

Room Number <input type="text"/>	Description of Room <input type="text"/>	
Building <input type="text"/>		
Type <input type="text"/>		
Max Capacity: <input type="text" value="3"/> <input type="button" value="▲"/>	Guest Info	
Occupied? <input type="radio"/> No	Last Name <input type="text"/>	
<input type="radio"/> Yes	First Name <input type="text"/>	
Needs Cleaned? <input type="radio"/> No	Check-In <input type="text"/> <input type="button" value="▼"/>	
<input type="radio"/> Yes	Check-Out <input type="text"/> <input type="button" value="▼"/>	
<input checked="" type="checkbox"/> Active		

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This will also come up when you create a new record, only the contents will be blank.)

<input type="button" value="Back"/>	<input type="button" value="Edit"/>	<input type="button" value="Save"/>
-------------------------------------	-------------------------------------	-------------------------------------

## Browse Guest Vehicles

**Browse Guest Vehicles**

<input type="checkbox"/> Active <input type="checkbox"/> Inactive	Make <input type="button" value="▼"/>	Model <input type="button" value="▼"/>	Location <input type="button" value="▼"/>	License# <input type="text"/>	GuestName <input type="text"/>
Arrived	<input style="width: 40px; height: 20px; border: 1px solid black;" type="text"/> / <input type="button" value="Calendar"/>	Time <input style="width: 100px; height: 20px; border: 1px solid black;" type="text"/>			
Departed	<input style="width: 40px; height: 20px; border: 1px solid black;" type="text"/> / <input type="button" value="Calendar"/>	Time <input style="width: 100px; height: 20px; border: 1px solid black;" type="text"/>	<input style="border: 1px solid black; padding: 2px 10px;" type="button" value="Search"/>		
GuestName	License	Make	Model	Color	Location/Parking #

### Read Guest Vehicle

<p>Ticket # <input type="text" value="545151"/></p> <p>License Plate <input type="text" value="1314xdf"/></p> <p>Make <input type="text" value="Chevrolet"/></p> <p>Model <input type="text" value="Impala"/></p> <p>Color <input type="text" value="White"/></p>	<p><b>Car Description:</b></p> <p><i>It is important to list any damages when vehicle is received to avoid fraud claims when given possession.</i></p>
<p><b>Dropped Off</b></p> <p>Date Arrived <input type="text" value="04/07/2020"/> </p> <p>Time Arrived <input type="text" value="10:22A"/></p> <p>Employee <input type="text" value="Mike Wozowski"/></p> <p>Location <input type="text" value="Nielsen Hall"/></p> <p>Parking # <input type="text" value="3"/></p>	
<p><b>Picked Up</b></p> <p>Date Departed <input type="text" value=" / "/> </p> <p>Time Departed <input type="text"/></p> <p>Employee <input type="text"/></p>	
<input checked="" type="checkbox"/> Active	
<input type="button" value="Back"/> <input type="button" value="Edit"/> <input type="button" value="Save"/>	

### Update Room

<p>Room Number <input type="text" value="4432"/></p> <p>Building <input type="text" value="Nielsen Hall"/></p> <p>Type <input type="text" value="Double Queen"/></p> <p>Max Capacity: <input type="text" value="3"/></p> <p>Occupied? <input type="radio"/> No <input checked="" type="radio"/> Yes</p> <p>Needs Cleaned? <input checked="" type="radio"/> No <input type="radio"/> Yes</p>	<p>Description of Room</p> <p>Guest Info</p> <p>Last Name <input type="text" value="Smith"/></p> <p>First Name <input type="text" value="John"/></p> <p>Check-In <input type="text" value="1/1/2018"/> </p> <p>Check-Out <input type="text" value="1/30/2018"/> </p>
<input type="button" value="Back"/> <input type="button" value="Edit"/> <input type="button" value="Save"/>	

### Update Guest Vehicle

Ticket #	454554
License Plate	1314xdf
Make	Chevrolet
Model	Impala
Color	White

#### Car Description:

*It is important to list any damages when vehicle is received to avoid fraud claims when given possession.*

#### Dropped Off

Date Arrived	04/07/2020	
Time Arrived	10:22A	
Employee	Mike Wozowski	
Location	Nielsen Hall	
Parking #	3	

#### Picked Up

Date Departed	/ /	
Time Departed		
Employee		

Active

[Back](#)

[Edit](#)

[Save](#)

### Update Shuttle

Shuttle VIN	01234567890123456	Shuttle Description:
License Plate	123 XYZ	This is a very apt description of this vehicle.
Shuttle Type	Autobus	
Shuttle Color	Yellow	
Shuttle Manufacturer	General Atomics	
Shuttle Name:	Big Battle Bus	<input type="checkbox"/> In Use <input type="checkbox"/> In Maintenance
Seats:	80	
Mileage	45600	
Driver Name	Tim Dalton	
Destination	Candy Shop	
Date Arrived	01/01/199	
<input checked="" type="checkbox"/> Active		
<a href="#">Back</a>	<a href="#">Edit</a>	<a href="#">Save</a>

### Deactivate Confirmation

Ticket#	<input type="text"/>	License	<input type="text"/>
Make	<input type="text"/>	Model	<input type="text"/>

Are you sure you would like to  
deactivate this guest vehicle?

[Cancel](#)[Deactivate](#)

### Deactivate Confirmation

Shuttle VIN

License Plate

Are you sure you would like to  
deactivate this shuttle?

### Deactivate Confirmation

Room#

Building

Type

Capacity

Are you sure you would like to  
deactivate this room?

**Deactivation Confirmation**

Name	<input type="text"/>	Email	<input type="text"/>
Phone	<input type="text"/>	Address	<input type="text"/>

Are you sure you would like to  
deactivate this appointment?

**Cancel**      **Deactivate**

Personnel Management

**New Employee Role**

Role Name	<input type="text"/>	Date:	<input type="text"/> / <input type="text"/> / <input type="text"/>	Description	<input type="text"/>
-----------	----------------------	-------	--	-------------	----------------------

**Cancel**      **Save**

**Create Schedule**

Day	Week1	Week2	Week3	Week4
Sunday	xx/01	xx/08	xx/15	xx/22
Monday	xx/02	xx/09	xx/16	xx/23
Tuesday	xx/03	xx/10	xx/17	xx/24
Wednesday	xx/04	xx/11	xx/18	xx/25
Thursday	xx/05	xx/12	xx/19	xx/26
Friday	xx/06	xx/13	xx/20	xx/27
Saturday	xx/07	xx/14	xx/21	xx/28

Month ▾

Double click on the data grid to bring up the record

---

Date <input type="text"/> 	<input style="border: none; border-radius: 50%; width: 20px; height: 20px; background-color: #ccc; color: black; font-size: 10px; padding: 0; margin-left: 10px;" type="button" value="Add Employee"/>	
Employee <input type="text"/>	Start Time <input type="text"/>	End Time <input type="text"/>
Employee <input type="text"/>	Start Time <input type="text"/>	End Time <input type="text"/>

**View Employee Roles**

search input

Role Name \_\_\_\_\_ Description \_\_\_\_\_

Double click on the data grid to bring up the record

Cancel	Add	Update	Delete
--------	-----	--------	--------

**Read Schedule**

Day	Week1	Week2	Week3	Week4
Sunday	xx/01	xx/08	xx/15	xx/22
Monday	xx/02	xx/09	xx/16	xx/23
Tuesday	xx/03	xx/10	xx/17	xx/24
Wednesday	xx/04	xx/11	xx/18	xx/25
Thursday	xx/05	xx/12	xx/19	xx/26
Friday	xx/06	xx/13	xx/20	xx/27
Saturday	xx/07	xx/14	xx/21	xx/28

Double click on the data grid to bring up the record

**Edit Line Detail**

---

Date	<input type="text"/>				
Employee	<input type="text"/>	Start Time	<input type="text"/>	End Time	<input type="text"/>
Employee	<input type="text"/>	Start Time	<input type="text"/>	End Time	<input type="text"/>

Add Employee

**Back** **Edit** **Save**

**Browse Employee Schedules**

Active  Inactive search input

Employee Name	Schedule
Michael Scott	M 8:00AM - 04:00PM T 8:00AM - 04:00PM

Cancel Add Read Delete

**List of Employee Roles**

Role ID	Role Name	Orginal Role	Role Start Date	Role Pay	Employee Name	Employee Id

Double click on the data grid to bring up the record.

**Add New Line**    **Edit Line Detail**

**Detail**

Role ID	12345678	Date Required	<input type="text"/>	<input type="checkbox"/> High Priority	Description of Role <input type="text"/>
Role Name	Some text	Employee Name	Some text		
Orginal Role	Some text	Employee ID	Some text		
Role Start Date	<input type="text"/>	Pay	<input type="button" value="3"/>		

**Save**

**Back**    **Edit**    **Save**

**Update Schedule**

Day	Week1	Week2	Week3	Week4
Sunday	xx/01	xx/08	xx/15	xx/22
Monday	xx/02	xx/09	xx/16	xx/23
Tuesday	xx/03	xx/10	xx/17	xx/24
Wednesday	xx/04	xx/11	xx/18	xx/25
Thursday	xx/05	xx/12	xx/19	xx/26
Friday	xx/06	xx/13	xx/20	xx/27
Saturday	xx/07	xx/14	xx/21	xx/28

Double click on the data grid to bring up the record

---

Date <input type="text"/>	<input type="button" value="Edit"/>	<input type="button" value="Save"/>
Employee <input type="text"/>	Start Time <input type="text"/>	End Time <input type="text"/>
Employee <input type="text"/>	Start Time <input type="text"/>	End Time <input type="text"/>
<input style="border: none; padding: 0 10px; border-radius: 50%; width: 20px; height: 20px; vertical-align: middle;" type="button" value="Add Employee"/>		

**Deactivation Confirmation**

Role Id	12345678	Date Required	<input type="text"/>
Role Name	Some text	orginal role	Some text

Are you sure you would like to deactivate \_\_\_?

### Deactivate/Delete Confirmation

Field ID:  Field:   
Field:  Field:

Are you sure you would like to deactivate \_\_\_?

### Dependent - CreateDependentDetails

ID Field

Name

Age

Gender

Type

Breed

Checked In  
Building  Room

Guest Contact 1  
Name   
Phone   
Email

Guest Contact 2  
Name   
Phone   
Email

**Dependents**

Checked In  Include Deactivated

ID	Name	Type	Gender	Breed	Building	Contact Name
----	------	------	--------	-------	----------	--------------

Double click on the data grid to bring up the record

**Cancel** **Add** **Update** **Delete**

Dependent - ViewDependentDetails

ID Field 12345678

Checked In

Building

Room 1123

Name Jim Bob

Age 7

Gender

Guest Contact 1

Name Dad Bob

Type

Phone 319-555-2112

Breed

Email dadbob@email.com

Guest Contact 2

Name

Phone

Email

Back

Edit

Save

Dependent - UpdateDependentDetails

ID Field

Name

Age

Gender

Type

Breed

Checked In  
Building   
Room

Guest Contact 1

Name   
Phone   
Email

Guest Contact 2

Name   
Phone   
Email

New Record

### Client Information

Name

Phone

Address

City

Zip Code

State

Email

### Tell us about your pet

Pet name

Service  ▾

Pet Species  ▾

Date:  

**List of Appointments**

App ID	Name	Email	Phone	Address	Pet Name	Service

Double click on the data grid to bring up the record

**Create new**      **Edit**

**Detail**

Appt ID	<input type="text"/>	Date	<input type="text"/>	Comments
Name	<input type="text"/>	Service	<input type="button" value="Follow Up"/>	
Email	<input type="text"/>	Duration	<input type="button" value="1 hr"/>	
Phone	<input type="text"/>	Gender	<input type="button" value="Male"/>	<input type="button" value="Female"/>
Pet Species	<input type="button" value="▼"/>	Status	<input type="button" value="Scheduled"/>	<input type="button" value="Save"/>
Gender	<input type="button" value="▼"/>			

**Cancel**      **Add**      **Update**      **Delete**

### [View Appointment](#)

Appt ID	<input type="text"/>	Services & Times	<input type="text"/> 1hour visit <input type="button" value="▼"/>
Client Name	<input type="text"/>		<input type="text"/> Morning(7:00 am - 8:00am) <input type="button" value="▼"/>
Address	<input type="text"/>		<input type="text"/> Notes
City	<input type="text"/>		
Zip Code	<input type="text"/>		
State	<input type="text"/>		
Phone	<input type="text"/>		
Email	<input type="text"/>		
Pet's Name	<input type="text"/>		
Pet Species	<input type="text"/> <input type="button" value="▼"/>		
Gender	<input type="text"/> <input type="button" value="▼"/>		
Start Date	<input type="text"/> <input type="button" value="Calendar"/>		
End Date	<input type="text"/> <input type="button" value="Calendar"/>		
<input type="button" value="Back"/>		<input type="button" value="Edit"/>	<input type="button" value="Save"/>

### Update Appointment

Appt ID	<input type="text"/>	Services & Times	<input type="text"/> 1hour visit <input type="text"/> Morning(7:00 am - 8:00am) <input type="text"/> Notes
Client Name	<input type="text"/>		
Address	<input type="text"/>		
City	<input type="text"/>		
Zip Code	<input type="text"/>		
State	<input type="text"/>		
Phone	<input type="text"/>		
Email	<input type="text"/>		
Pet's Name	<input type="text"/>		
Pet Species	<input type="text"/> ▾		
Gender	<input type="text"/> ▾		
Start Date	<input type="text"/>	<input type="button" value="Save"/>	<input type="button" value="Delete Record?"/>
End Date	<input type="text"/>		

[Back to List](#)

### Deactivation Confirmation

Name	<input type="text"/>	Email	<input type="text"/>
Phone	<input type="text"/>	Address	<input type="text"/>

Are you sure you would like to deactivate this appointment?

New Record

### Client Information

Name

Phone

Address

City

Zip Code

State

Email

### Tell us about your pet

Pet name

Service  ▾

Pet Species  ▾

Date:  /  

### View Appointment

Appt ID	<input type="text"/>	Services & Times	<input type="text" value="1hour visit"/> <input type="text" value="Morning(7:00 am - 8:00am)"/>
Client Name	<input type="text"/>		
Address	<input type="text"/>		
City	<input type="text"/>		
Zip Code	<input type="text"/>		
State	<input type="text"/>		
Phone	<input type="text"/>		
Email	<input type="text"/>		
Pet's Name	<input type="text"/>		
Pet Species	<input type="text"/>		
Gender	<input type="text"/>		
Start Date	<input type="text"/> 		
End Date	<input type="text"/> 		
<input type="button" value="Back"/>		<input type="button" value="Edit"/>	<input type="button" value="Save"/>

### Update Appointment

Appt ID	<input type="text"/>	Services & Times	<input type="text"/> 1 hour visit <input type="text"/> Morning(7:00 am - 8:00am) <input type="text"/> Notes
Client Name	<input type="text"/>		
Address	<input type="text"/>		
City	<input type="text"/>		
Zip Code	<input type="text"/>		
State	<input type="text"/>		
Phone	<input type="text"/>		
Email	<input type="text"/>		
Pet's Name	<input type="text"/>		
Pet Species	<input type="text"/> ▾		
Gender	<input type="text"/> ▾		
Start Date	<input type="text"/>		<input type="button" value="Save"/>
End Date	<input type="text"/>		<input type="button" value="Delete Record?"/>
<input type="button" value="Back to List"/>			

## Event Operations

Event Detail Edit / Create New Event

Event Name	<input type="text" value="Name"/>	Type of Event	<input type="button" value="Event Type ▾"/>
Host	<input type="button" value="Host ▾"/>	Requestor	
Requestor			
Event Date:	<input type="text" value="/ /"/> <input type="button" value="Calendar"/>	Description of Event	
Location	<input type="button" value="Locations ▾"/>		
Catering	<input type="button" value="Catering ▾"/> <input type="button" value="Details"/>		
<input type="checkbox"/> Pets Allowed			
<input type="checkbox"/> Children Allowed			
<input type="checkbox"/> Talent Performing			
<input type="button" value="Add RSVP List"/>			
<input type="button" value="Assign Employees to Event"/>			
<input type="button" value="Cancel"/>	<input type="button" value="Save"/>		

**View/Edit/Delete Events**

Open       Completed

Event Name	Date	Location	Requestor	Host	Description

Double click on the data grid to bring up the record

**Cancel**      **Edit**      **Save**

**New Event Request**

Requestor:	<input type="text" value="HostName"/>	Description	
Event Name:	<input type="text"/>		
Location:	<input type="text"/>		
Event Date:	<input type="text"/> / <input type="text"/>		
Event Type:	<input type="text"/>		
<input type="checkbox"/> Talent?	<input type="text" value="Talent"/>		
 		<input type="checkbox"/> Pets Allowed?	
		<input type="checkbox"/> Children Allowed?	Approved?
		<input type="checkbox"/> Catering Request?	<input type="checkbox"/> Yes
		<input type="button" value="Submit Cater Request"/>	<input type="checkbox"/> No
			<input checked="" type="checkbox"/> Waitlist

**Cancel**      **Create**

**Create Catering Request**

Event Name

Host

Requestor

Event Date:

Location

Catering

Food type

Catering details

Cancel

Create

**Create Advertising event**

Advertisement Header:

Advertisement Motto:

Start Date:

End Date:

Area Advertised:

Advertised by:

Contact Info Mentioned:

Advertised On:

Advertisement Details

Cancel

save

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

### New Performance

Performance Name:

Event Selector:  

First Name:

Event Date:  

Last Name:

Talent Name:

Talent Description:

**New Setup List**

Date Created	<input type="text" value="04/04/1992"/>	Date Modified	<input type="text" value=" / /"/>	<input type="button" value="Calendar"/>
ID	<input type="text" value="515185181"/>			
Tasks	<input type="text"/>			
Add Task	<input type="button" value="+"/>			

<input type="button" value="Cancel"/>		<input type="button" value="Save"/>
---------------------------------------	--	-------------------------------------

**Read Advertising event**

Advertisement Header:	<input type="text"/>	Advertisement Details	<p>(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This will also come up when you create a new record, only the contents will be blank.)</p>
Advertisement Motto:	<input type="text"/>		
Start Date:	<input type="text"/> / <input type="button" value="Calendar"/>		
End Date:	<input type="text"/> / <input type="button" value="Calendar"/>		
Area Advertised:	<input type="button" value="Locations"/>		
Advertised by:	<input type="text" value="EmployeeName"/>		
Contact Info Mentioned:	<input type="text"/>		
Advertised On:	<input type="text" value="Billboards, tv, newspaper, etc"/>		

<input type="button" value="Cancel"/>		<input type="button" value="Update"/>
---------------------------------------	--	---------------------------------------

Read/Update Event Request

Requestor:	<input type="text" value="HostName"/>	Description			
Event Name:	<input type="text"/>				
Location:	<input type="text"/>				
Event Date:	<input type="text"/> / <input type="button" value="Calendar"/>				
Event Type:	<input type="text"/>				
<input type="checkbox"/> Talent?	<input type="text" value="Talent"/>				
<input type="checkbox"/> Pets Allowed?			Approved?	<input type="checkbox"/> Yes	
<input type="checkbox"/> Children Allowed?			<input type="checkbox"/> No		
<input type="checkbox"/> Catering Request?			<input checked="" type="checkbox"/> Waitlist		
<input type="button" value="Submit Cater Request"/>					

Read Event Details

event ID: 12345678	Date Required	<input type="text"/> <input type="button" value="Calendar"/>	Event Description
event customer	<input type="text"/>	<input type="checkbox"/> High Priority	
Event Name:	<input type="text" value="Some text"/>		
Event Date	<input type="text"/> <input type="button" value="Calendar"/>	Number of guests	<input type="text" value="3"/> <input type="button" value="Up"/> <input type="button" value="Down"/>
Event	<input type="text"/>	Number of Employees	<input type="text" value="3"/> <input type="button" value="Up"/> <input type="button" value="Down"/>
Demographic	<input type="checkbox"/> Kids <input type="checkbox"/> Adults <input type="checkbox"/> Young Adults <input type="checkbox"/> Teens <input type="checkbox"/> Seniors <input type="checkbox"/> Infants <input type="checkbox"/> Pets		
	Event Open to: <input type="radio"/> Open to all <input type="radio"/> Open to invited guests <input type="radio"/> Open to invited guest and emp <input type="radio"/> employee only		
<input type="button" value="Cancel"/>	<input type="button" value="Save"/>		
<input type="button" value="Back to List"/>			

### Read Catering List

Date Created

Date Modified

ID	<input type="text" value="51515185181"/>
Tasks	<input type="text" value="Buy Nice Tablecloths"/>
	<input type="text" value="Prepare Table with Tablecloth"/>
	<input type="text" value="Clean Tablecloth After Kid Throws Up On It"/>

### Read Performance

Performance Name:

Event Selector:  

First Name:

Event Date:  

Last Name:

Talent Name:

Talent Description:

View/Edit/Delete Event Requests

Waitlist     Talent

HostName	EventName	Date	Children?	Pets?	Description

Double click on the data grid to bring up the record

Search Bar

List of Recipes

search category	search input				
HostName	EventName	Date	Children?	Pets?	Description

Cancel      Add      Read      Delete

**Browse Performances**

Active    Deactive

search

Performance Name	LastName	Talent	Event	EventDate	Talent Desc

Double click on the data grid to bring up the record

Browse Advertising Events

Active  Inactive     Filter 1  Filter 2  Filter 3  Filter 4

AdvertisementHeader — StartDate — EndDate — Type — Location — Description

AdvertisementHeader	StartDate	EndDate	Type	Location	Description

Browse Setup List

search

ID Field	Date Created	Date Modified
----------	--------------	---------------

Cancel      Delete      Read/Update      Add

Browse Catering Requests

<input type="checkbox"/> Active	<input type="checkbox"/> Inactive	<input type="radio"/> Open	<input type="radio"/> Completed	search input	
Req ID	Date	Location	Requestor	Host	Description

### Read Catering Request

Event Name

Type of Event

Host  ▾

Catering details

Requestor

Event Date:  

Location  ▾

Catering  ▾

Food type  ▾

Pets Allowed

Children Allowed

Talent Performing

### Update Catering Request

Event Name	<input type="text" value="Name"/>	Type of Event	<input type="button" value="Event Type ▾"/>
Host	<input type="text" value="Host"/> <input type="button" value="▼"/>	Catering details	
Requestor	<input type="text" value="Requestor"/>		
Event Date:	<input type="text" value="/ /"/> <input type="button" value="Calendar"/>		
Location	<input type="text" value="Locations"/> <input type="button" value="▼"/>		
Catering	<input type="text" value="Catering"/> <input type="button" value="▼"/>		
Food type	<input type="text" value="Food Types"/> <input type="button" value="▼"/>		
<input type="checkbox"/> Pets Allowed <input type="checkbox"/> Children Allowed <input type="checkbox"/> Talent Performing			
<input type="button" value="Create"/>		<input type="button" value="Back to Event"/>	

### Update Advertising event

Advertisement Header:	<input type="text"/>	Advertisement Details	(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This will also come up when you create a new record, only the contents will be blank.)
Advertisement Motto:	<input type="text"/>		
Start Date:	<input type="text" value="/ /"/> <input type="button" value="Calendar"/>		
End Date:	<input type="text" value="/ /"/> <input type="button" value="Calendar"/>		
Area Advertised:	<input type="text" value="Locations"/> <input type="button" value="▼"/>		
Advertised by:	<input type="text" value="EmployeeName"/>		
Contact Info Mentioned:	<input type="text"/>		
Advertised On:	<input type="text" value="Billboards, tv, newspaper, etc"/> <input type="button" value="▼"/>		
<input type="button" value="Cancel"/>		<input type="button" value="Save"/>	

Read Performance

Performance Name:	<input type="text"/>	Event Selector:	<input type="button" value="ComboBox"/>
First Name:	<input type="text"/>	Event Date	<input type="text"/>
Last Name:	<input type="text"/>		
Talent Name:	<input type="text"/>	Talent Description:	<input type="text"/>
		<input checked="" type="checkbox"/> Active <input type="checkbox"/> Deactive	
<input type="button" value="Back"/>		<input type="button" value="Edit"/>	<input type="button" value="Save"/>

Employee xxxxxxx

Employee ID	12345678	Date Required	<input type="text"/>
First Name:	<input type="text"/>	Last Name:	<input type="text"/>
Event Some text		Role: Some text	
Event Date	<input type="text"/>	Age	<input type="button" value="3"/>
Field	<input type="button" value=""/>	Field	<input type="button" value="3"/>
Field	<input type="button" value=""/>	Field	Some text
Field	<input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option	Field	<input type="radio"/> option <input type="radio"/> option <input type="radio"/> option <input type="radio"/> option
Field	<input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option	Field	Some text
Field	<input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option <input type="checkbox"/> option	Field	Some text
		<input type="button" value="Save"/>	<input type="button" value="Delete Record?"/>
<input type="button" value="Back to List"/>			

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

Employee xxxxxxx

Employee ID 12345678

Date Required  

First Name:

Last Name:

High Priority

Event Some text

Role: Some text

Event Date  

Age  

### Catering Setup List

Date Created 04/04/1992

Date Modified 04/04/1992

ID

151815218818

Tasks

Buy Nice Tablecloths

Prepare Table with Tablecloth

Clean Tablecloth After Kid Throws Up On It

Add Task



Back

Save

Edit

### Deactivation Confirmation

Event Name

Host:

Requestor

Event Date  / / A small icon of a calendar or date picker.

Are you sure you would like to  
deactivate this catering  
request?

Cancel

Deactivate

**Deactivation Confirmation**

AdvertiserID:  StartDate:  / / 

Headline:  Type:  ▾

Are you sure you would like to deactivate this advertisement event?

**Cancel** **Deactivate**

**Deactivation Confirmation**

Performance Name  Talent Name

Event Name  Date  / / 

Are you sure you would like to deactivate this Performance?

**Cancel** **Deactivate**

**Deactivation Confirmation**

RequestID:  Requestor:

EventName:  Date:  /

Are you sure you would like to deactivate this event request?

**Deactivation Confirmation**

Event Name:  Name  Type of Event:  Event Type

Host:  Host  Event Date:  /

Are you sure you would like to deactivate   ?

**Deactivation Confirmation**

ID

Tasks

Are you sure you would like to deactivate this Setup List?

**New Reservation**

Name <input type="text" value="Reserver Name"/>	Reservation Details
Receptionist <input type="text" value="Name"/>	<input type="text" value="Description"/>
Date Reserved <input type="text"/>	
Date Departure <input type="text"/>	
Group Size <input type="text" value="3"/>	
# of Kids <input type="text" value="3"/>	
# of Pets <input type="text" value="3"/>	
Room Type <input type="text" value="Room Type"/>	

**Read Reservation**

Name	<input type="text" value="Reserver Name"/>	Reservation Details	
Receptionist	<input type="text" value="Name"/>	<input type="text" value="Description"/>	
Date Reserved	<input type="text"/>		
Date Departure	<input type="text"/>		
Group Size	<input type="text" value="3"/>		
# of Kids	<input type="text" value="3"/>		
# of Pets	<input type="text" value="3"/>		
Room Type	<input type="text" value="Room Type"/>		

**Receptionist - View List of Reservations**

GuestName	ArrivalDate	DepartureDate	#OfGuests	Receptionist	Details

Double click on the data grid to bring up the record

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

**Update Reservation**

<p>Name <input type="text" value="Reserver Name"/></p> <p>Receptionist <input type="text" value="Name"/></p> <p>Date Reserved <input type="text"/> </p> <p>Date Departure <input type="text"/> </p> <p>Group Size <input type="text" value="3"/> </p> <p># of Kids <input type="text" value="3"/> </p> <p># of Pets <input type="text" value="3"/> </p> <p>Room Type <input type="text" value="Room Type"/> </p>	<p><b>Reservation Details</b></p> <p>Description <input type="text"/></p>
--	---

(This is the form that comes up when you choose "edit" on something from the list view. This is where all details of a record are shown AND where you edit the details of records if you have permissions. This Will also come up when you create a new record, only the contents will be blank.)

**Deactivate Confirmation**

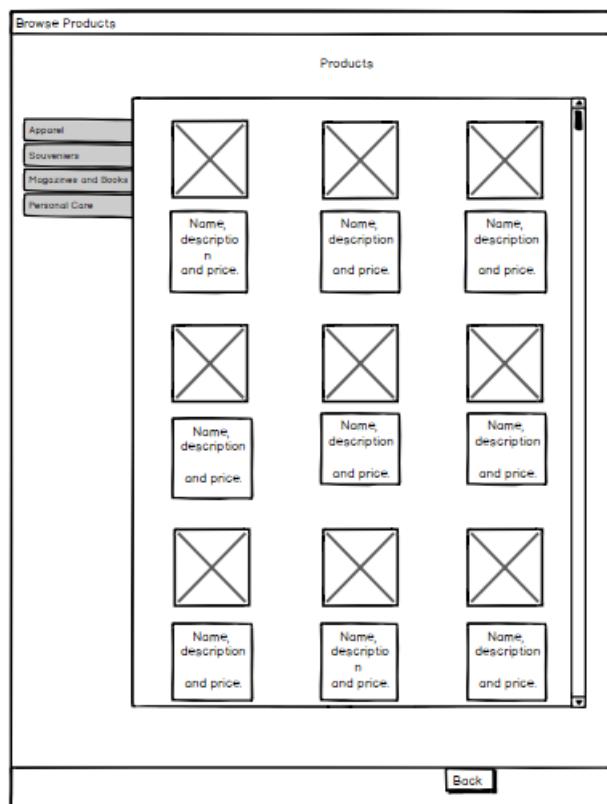
<p>Name <input type="text"/></p> <p>Date Reserved <input type="text"/></p>
--

**Are you sure you would like to deactivate this reservation?**

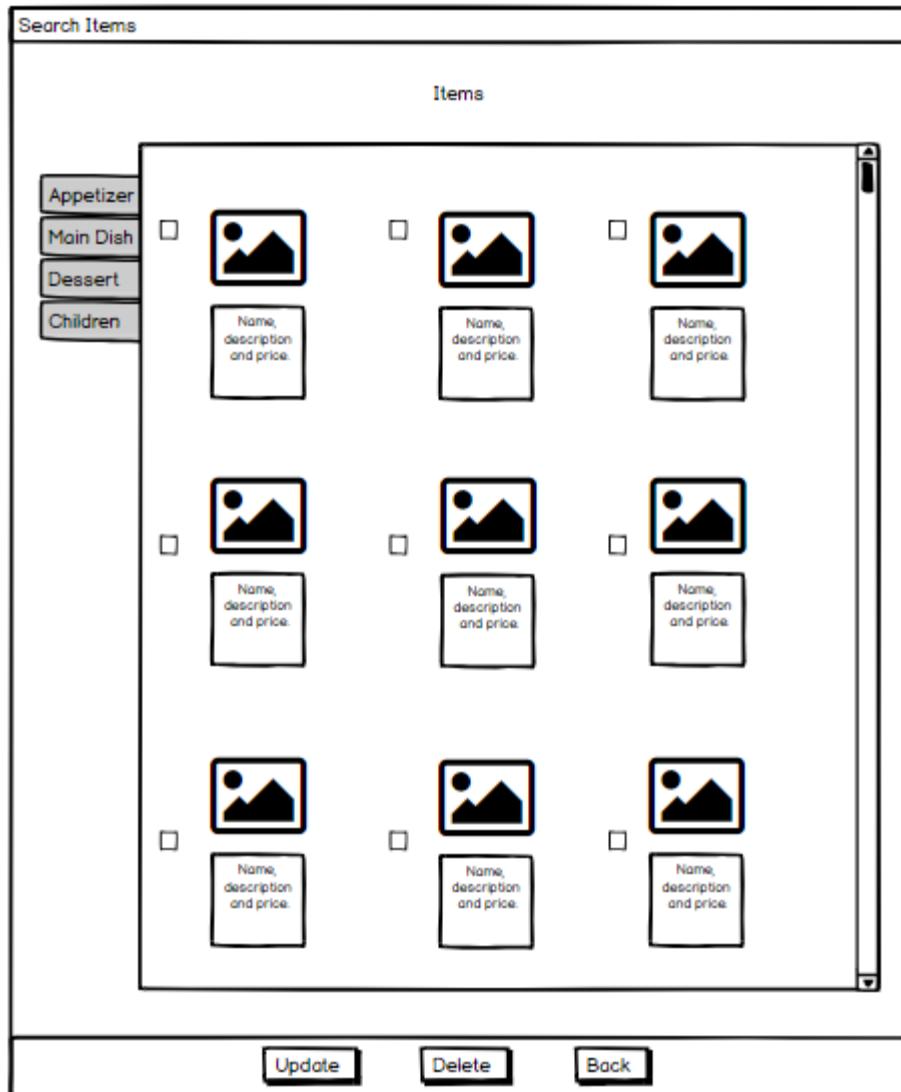
## External

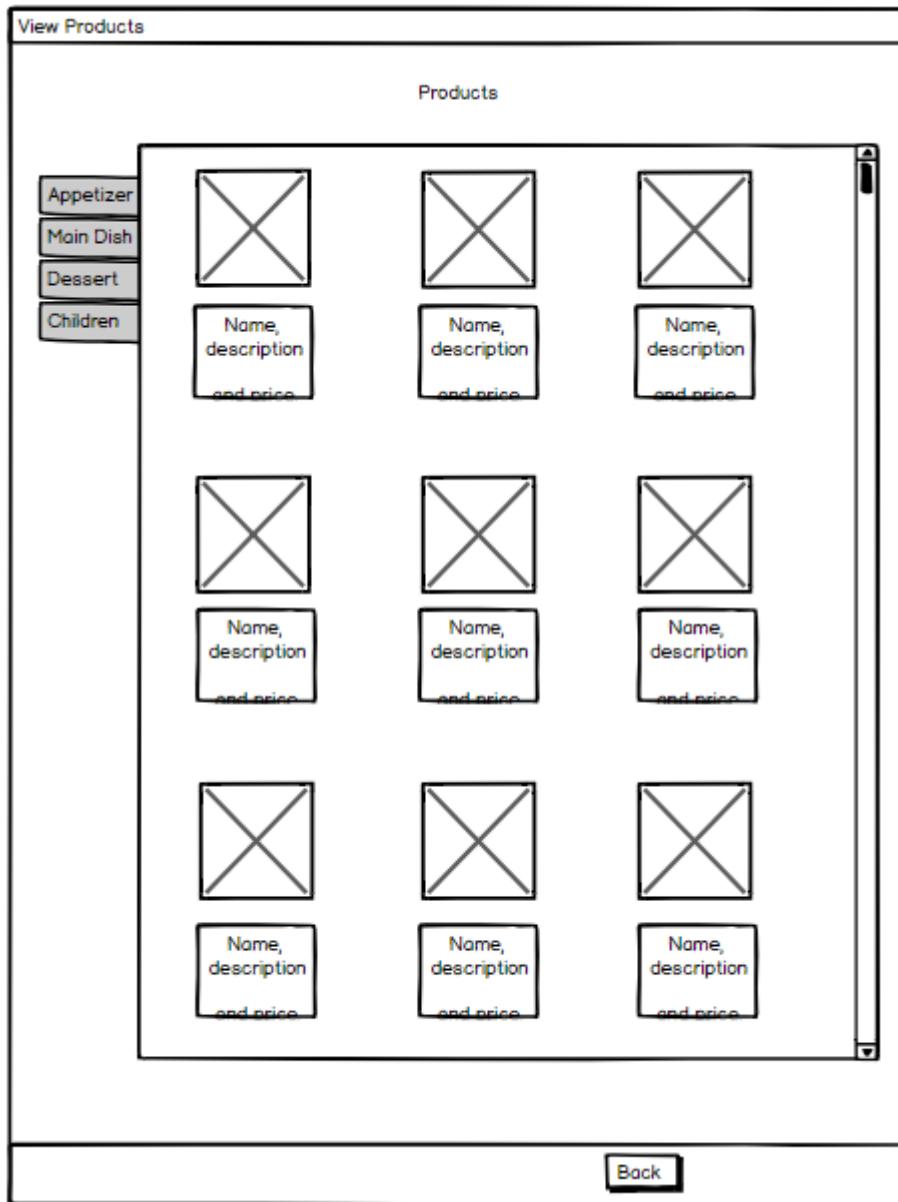
The external interfaces are where web users will interact with the individual functions of the solution. As with the internal interfaces, the design intent is to make the screens have a common look and feel and easy to use.

## Supply Operations



ReadCart	
Cart	
Product A	Price
	\$\$\$\$
<a href="#">Delete</a> <a href="#">Details</a>	
Product A	\$\$\$\$
<a href="#">Delete</a> <a href="#">Details</a>	
Subtotal	\$\$\$\$
<a href="#">Checkout</a>	





## Guest Operations

Create Account

Account Information 

Name	<input type="text" value="XXXXXXXXXX"/>	<input type="checkbox"/> Receive Event Announcements
Last Name	<input type="text" value="XXXXXXXXXX"/>	<input type="checkbox"/> Receive Resort Deals
E-mail	<input type="text" value="XXXXXXXXXX"/>	
Date of Birth	<input type="text" value="XX-XX-XX"/> 	
Phone number	<input type="text" value="XXXXXXXXXX"/>	
Address	<input type="text" value="XXXXXXXXXX"/>	
City	<input type="text" value="XXXXXXXXXX"/>	
Zip code	<input type="text" value="XXXXXXXXXX"/>	

Window Name

## ***REVIEW EVENT***

Title of Event

Date / /

Provided by system

Rating



Name

Optional

E-mail

Optional

Personalized message

Submit

Cancel

UpdatedDeleteAccount

Account Information ?

Name   Receive Event Announcements

Last Name   Receive Resort Deals

E-mail

Date of Birth  

Phone number

Address

City

Zip code  Date Created

Window Name

## REVIEW EVENT

Title of Event

Date / /  Provided by system

Rating

★ ★ ★ ★ ★

Name  Optional

E-mail  Optional

Personalized message

Hello, please sign in.

Username

[Forgot Username?](#)

Password

[Forgot Password?](#)

ReadAccount

Account Information 

Name   Receive Event Announcements

Last Name   Receive Resort Deals

E-mail 

Date of Birth  

Phone number 

Address 

City 

Zip code  Date Created 



*Product, Service or Event Name*Filters     1★     2★     3★ search

★★★★★	Comments on review

Back

Window Name

## REVIEW EVENT

Title of Event

Date / /

Provided by system

Rating



Name Optional

E-mail Optional

Personalized message

Update

Delete

Back

UpdatedAccount

Account Information 

Name   Receive Event Announcements

Last Name   Receive Resort Deals

E-mail 

Date of Birth  

Phone number 

Address 

City 

Zip code  Date Created 

Window Name

## REVIEW EVENT

Title of Event

Date  /  / Provided by system

Rating

Name  Optional

E-mail  Optional

Personalized message

Update  Delete  Back

Event Operations

CreateReservationRestaurantWeb

Event Name

Location  
Select a Restaurant

Date and Time of Reservation

Start date  End date

For how many will people is the reservation

List of people invited

Name  Email or PN  Add

Are pets allowed   
Are Kids allowed

Special Request

Window Name

# RSVP

Title of Event

Kindly respond by Date / /

Attend

Not Attending

Name of invited person

Food Menu ▾

Personalized message

Update

Delete

Back

Window Name

## Performances

Filters:

search

	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three

Back

Window Name

## Performances

Filters:

search

	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three
	<b>Event Name</b>	Details of the Event Item One Item Two Item Three

Back

Read Event

Event Name

Type of event

How many participants

Date and Time

Decorations   
List of decorations and details

Catering services   
List of decorations and details

Table skirt, cloth and napkins

Children

Pet Friendly

Special Request

Window Name

# RSVP

Title of Event

Kindly respond by Date / /

Attend       Not Attending

Name of invited person

Food Menu ▾

Personalized message

Update   Delete   Back

Schedule Booking		Update Booking	
Booking Details			
Room Type	<input type="button" value="Type"/>	Room Picture	
Details of Room	<input type="button" value="List of room details"/> Pet friendly No smoking Adults Child		
Date and Time	<input type="button" value="Arriving"/> <input type="button" value="Leaving"/>	<input type="button" value="Arriving"/> <input type="button" value="Leaving"/>	
Bed Choices	<input type="button" value="Bed Type"/>		
<input type="checkbox"/> Additional Bed <input type="checkbox"/> Minibar <input type="button" value="Special or Accessibility Request"/>			
<input type="button" value="Customer Details"/>			
First Name	<input type="text" value="Input Name"/>		
Last Name	<input type="text" value="Input Last Name"/>		
E-mail	<input type="text" value="Input E-mail"/>		
Phone number	<input type="text" value="Input phone number"/>		
<input type="button" value="Submit"/>		<input type="button" value="Cancel"/>	
<input type="button" value="Update"/>			
<input type="button" value="Cancel"/>			

Window Name

# RSVP

Title of Event

Kindly respond by Date / /

Attend

Not Attending

Name of invited person

Food Menu ▾

Personalized message

Submit

Back

Read Booking

Booking Details

Room Type      Room type selected

Details of Room      List of room details  
Pet friendly  
No smoking  
Adults  
Child

Room Picture

Date and Time      Arriving      Leaving

Bed Choice      Bed type selected

Additional Bed

Minibar

Special or Accessibility Request

Customer Details

First Name      Input Name

Last Name      Input Last Name

E-mail      Input E-mail

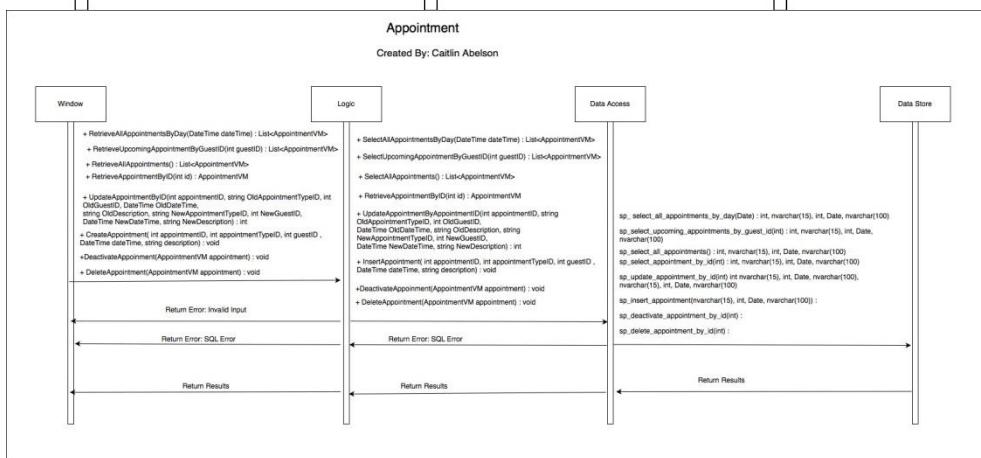
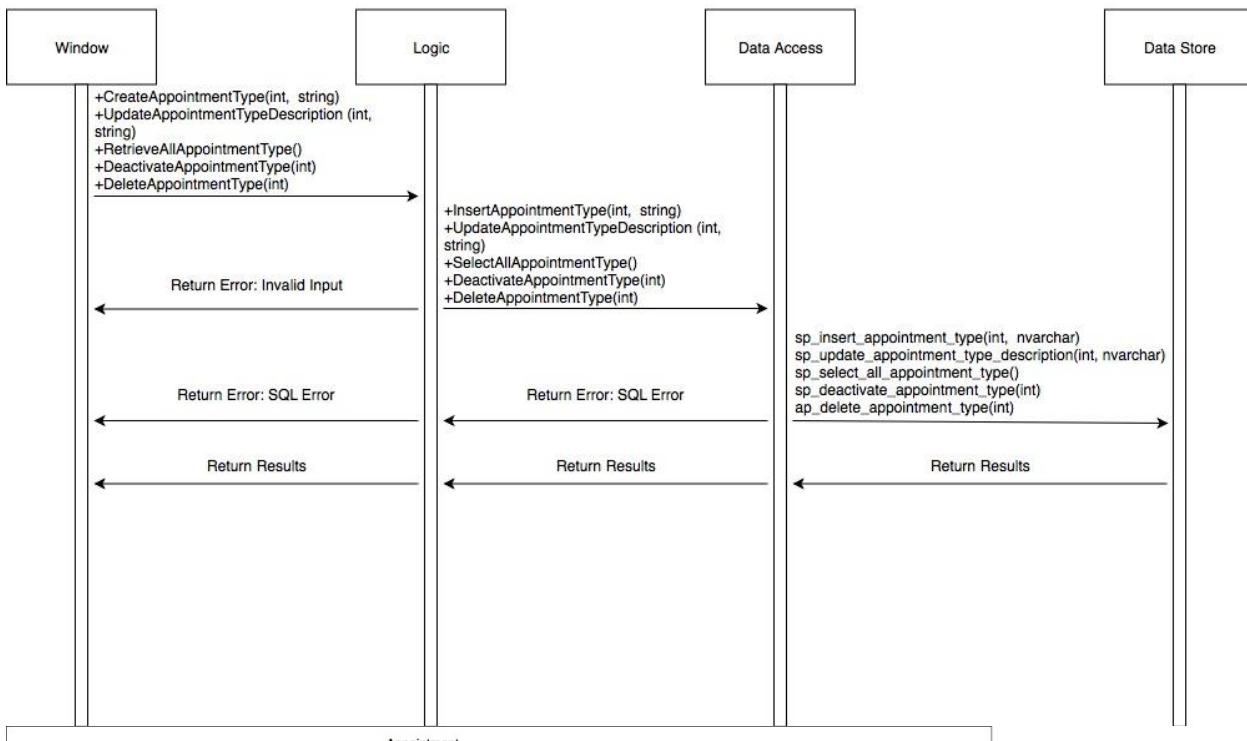
Phone number      Input phone number

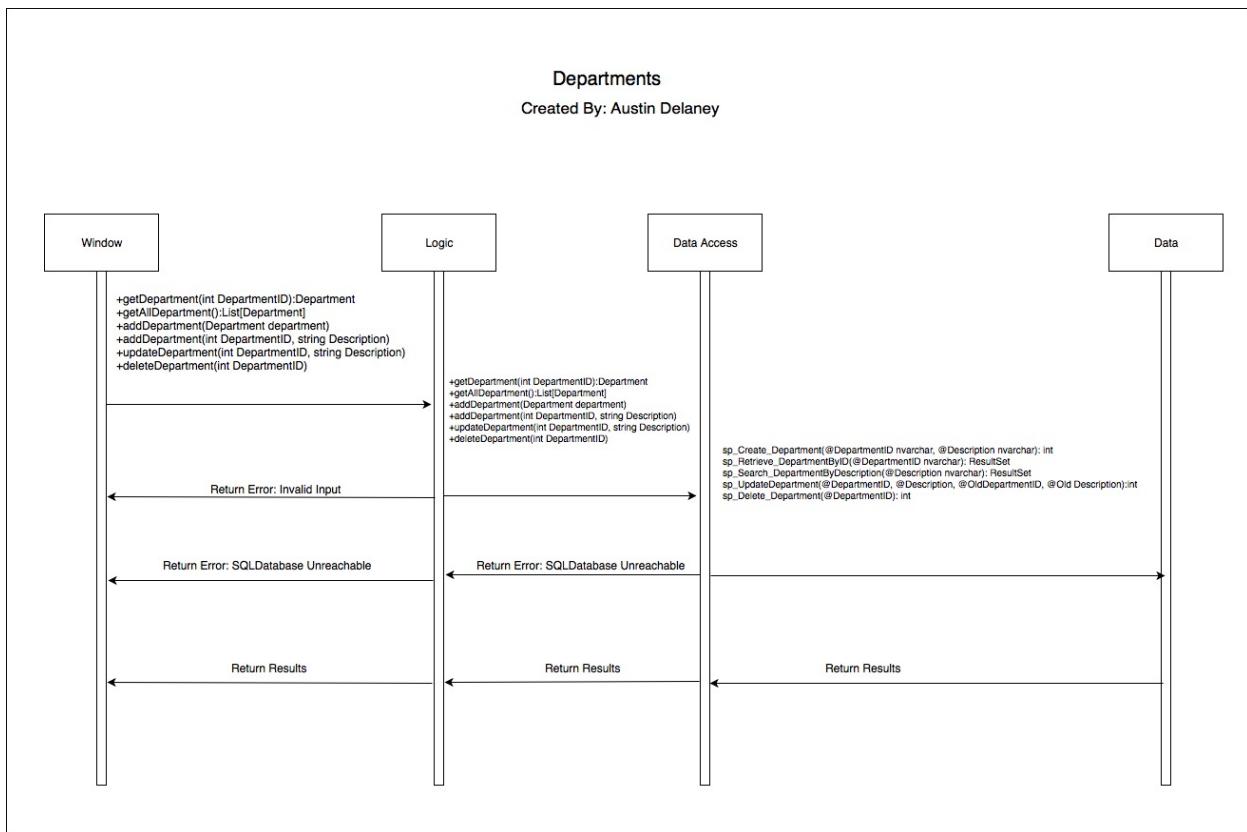
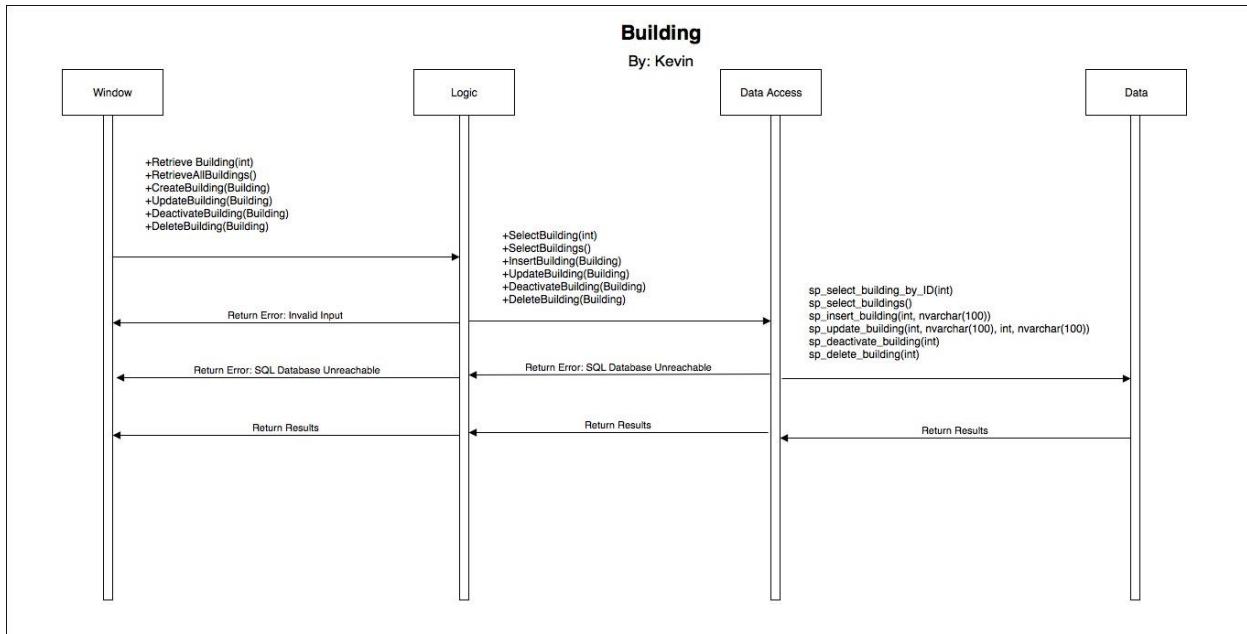
Update      Back

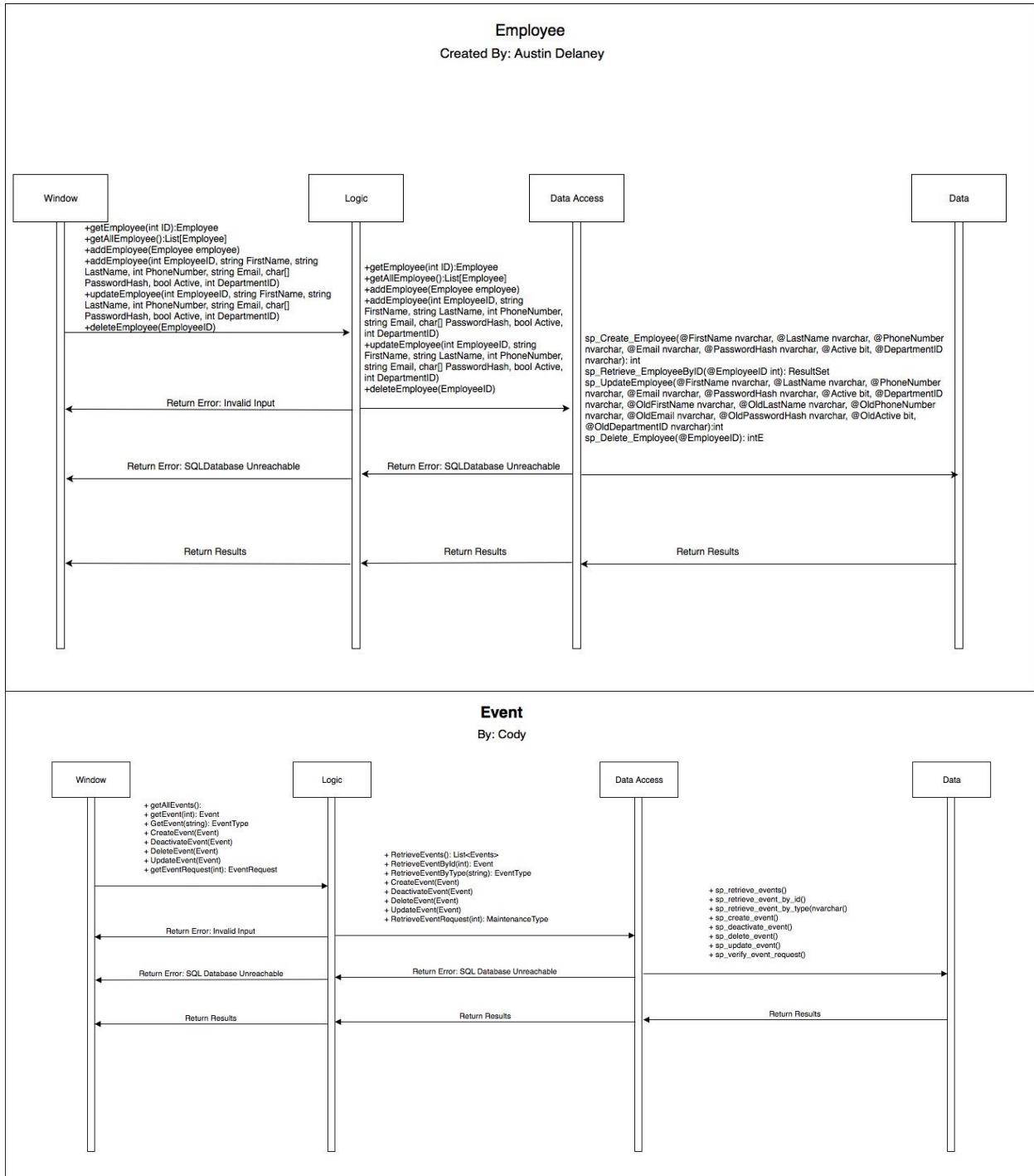
## Sequence Diagrams

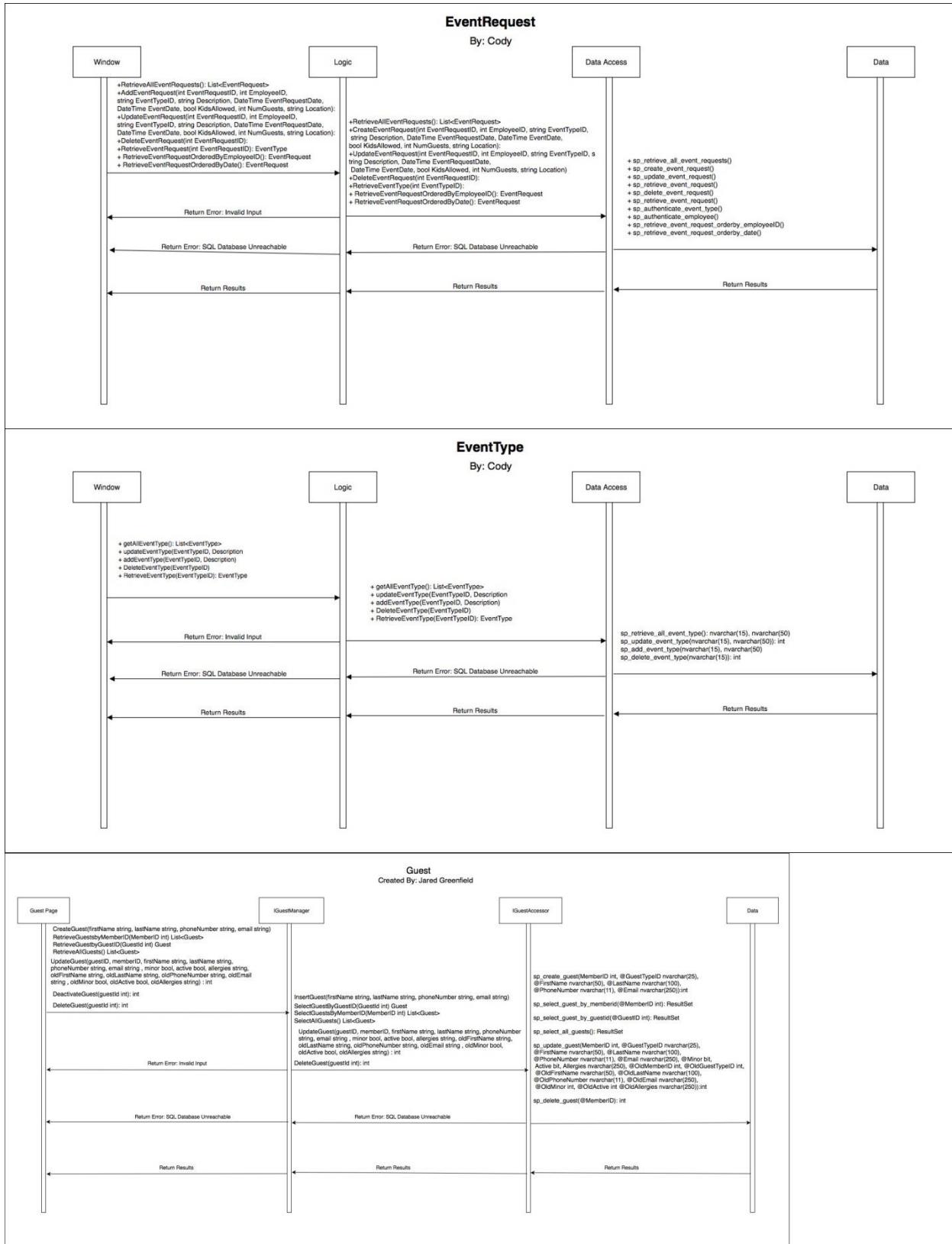
The sequence diagrams illustrate all available paths following a user's initial action. Each diagram represents a resort entity and specifies the actions available and the communication that occurs within the program. All expected results of a requested action, including errors, are shown in the sequence they would occur.

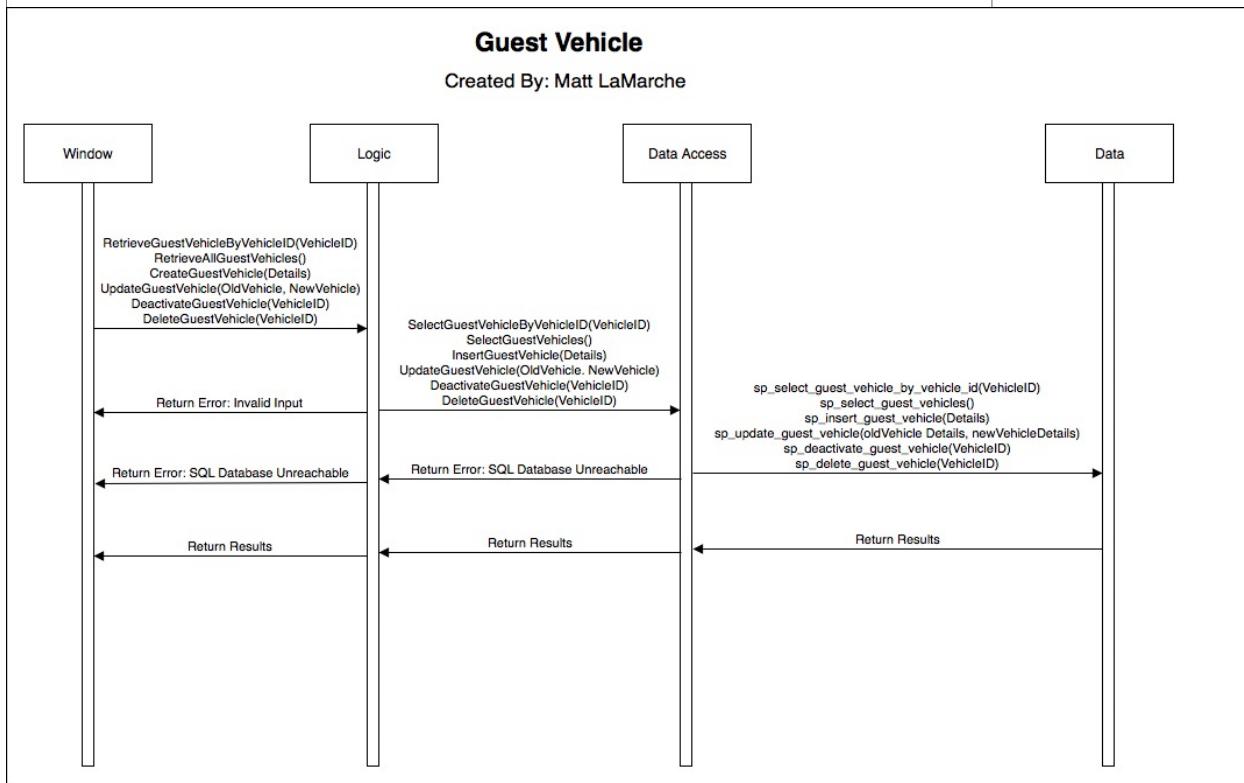
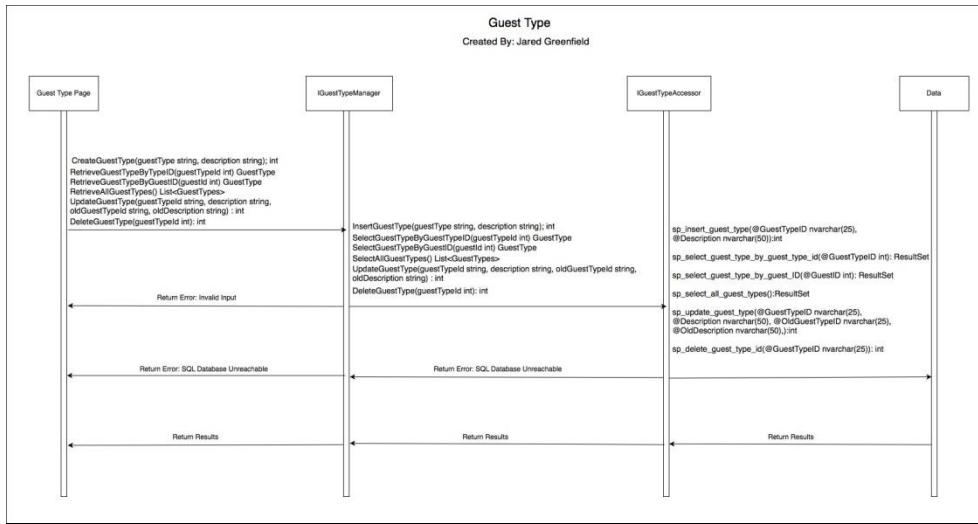
**Appointment Type**  
**Created By:** Jacob Miller  
**Edited By:** Jared Greenfield

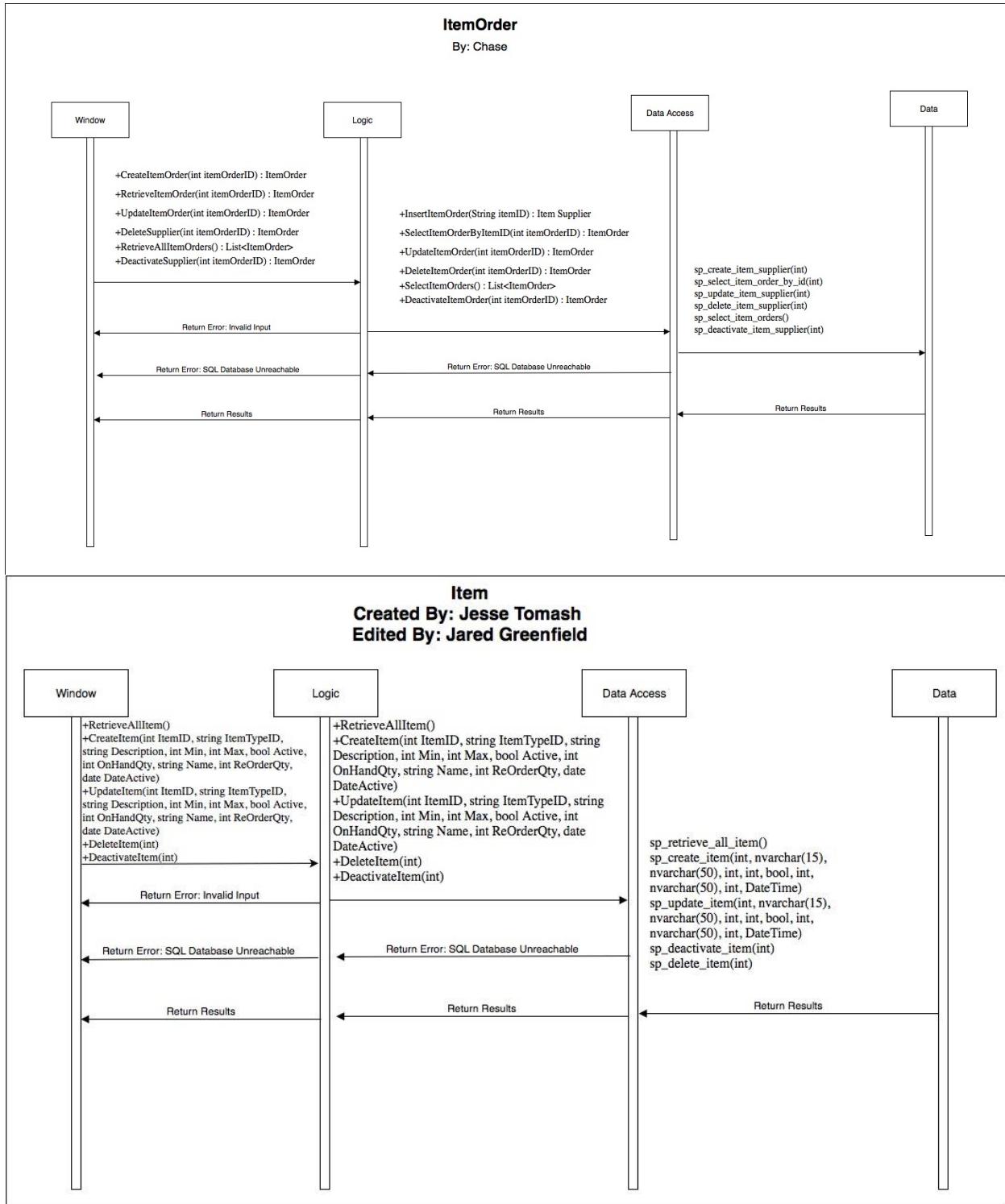




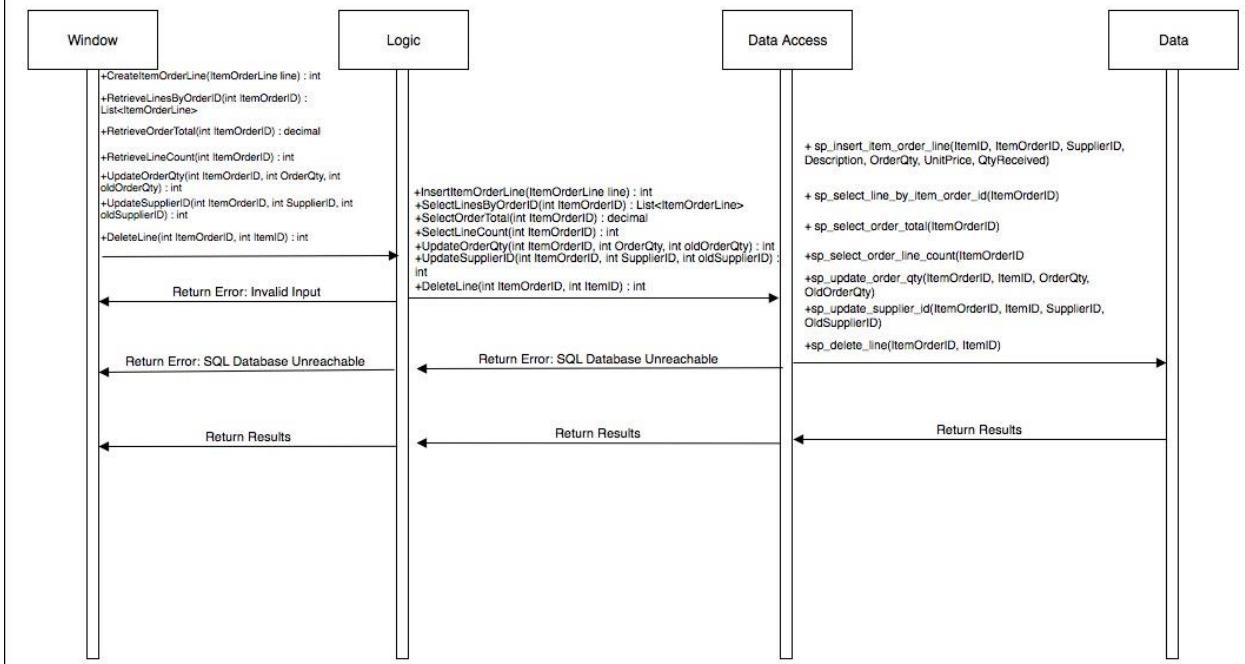






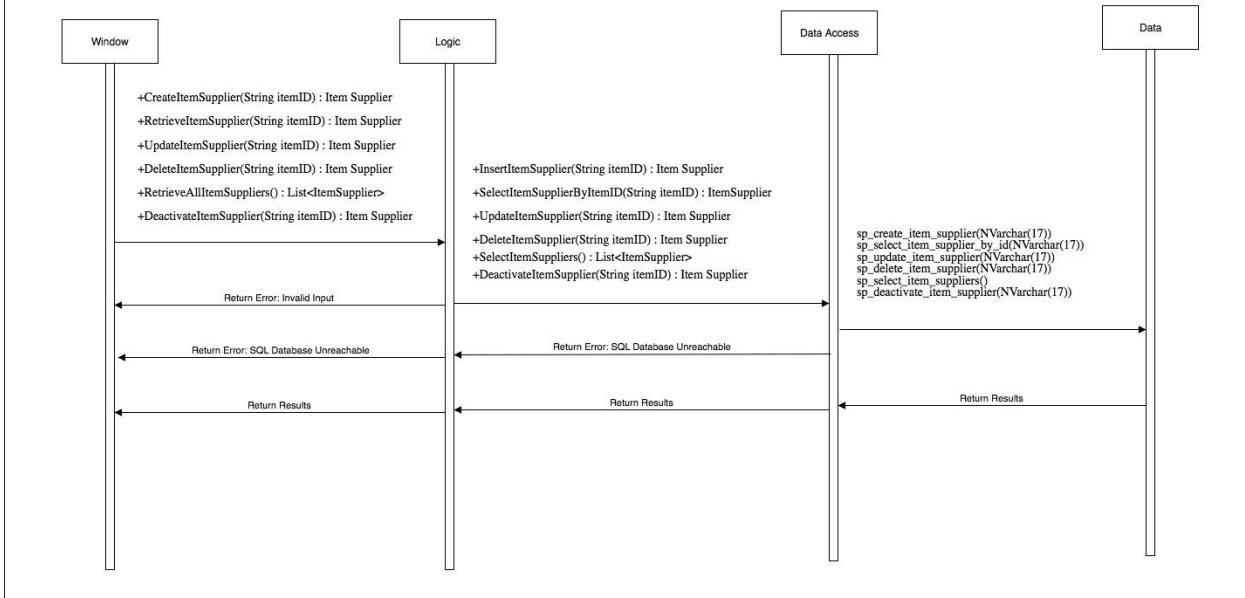


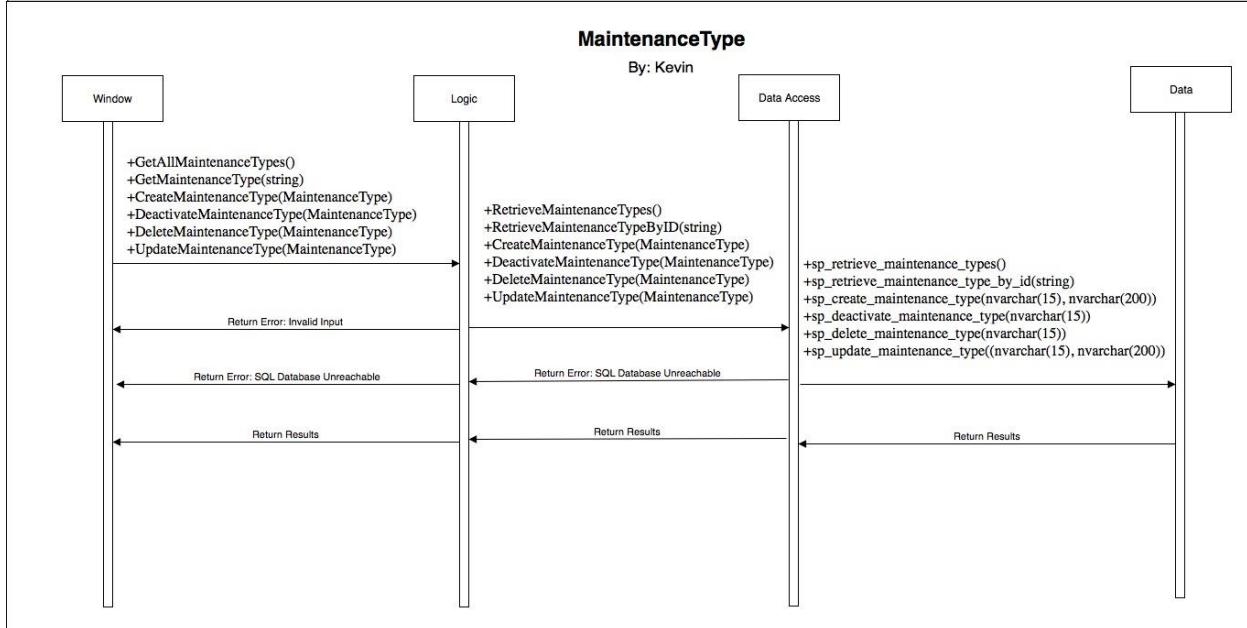
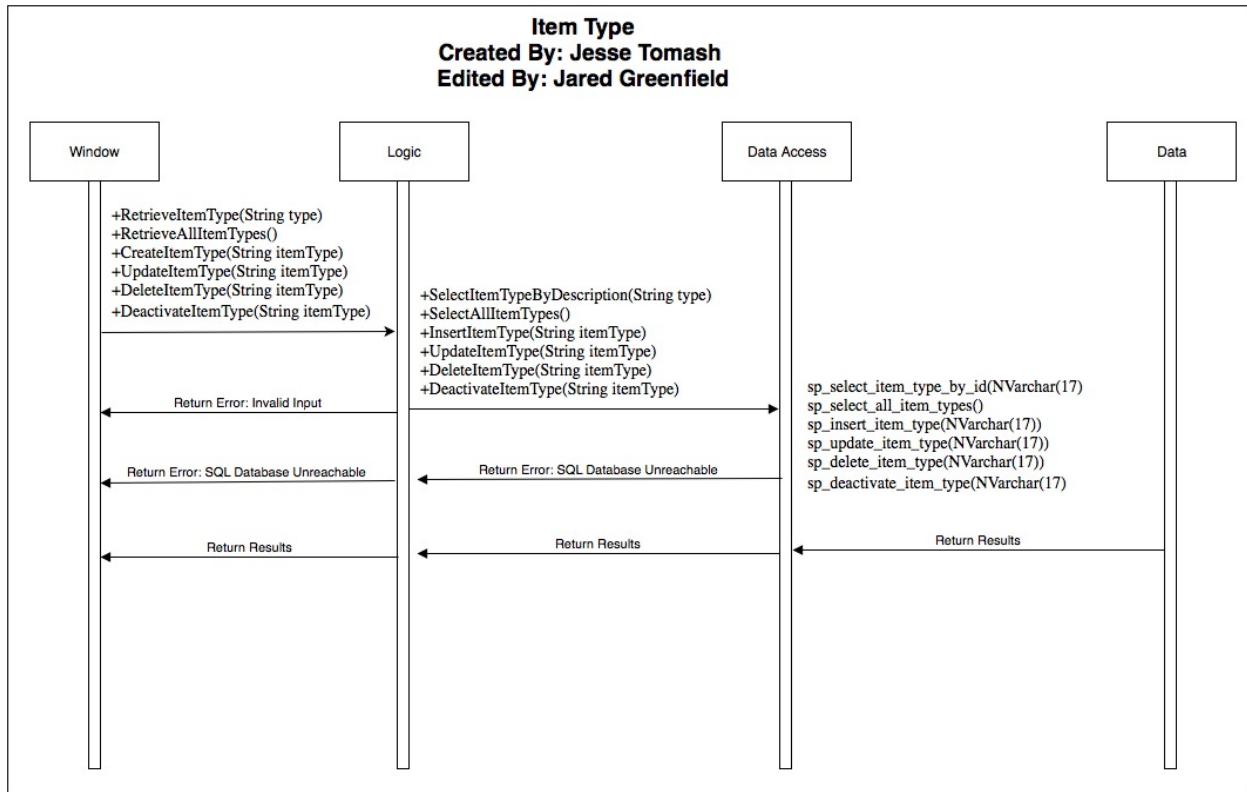
**Item Order Line**  
**Created By: Ramesh Adhikari**  
**Edited By: Jared Greenfield**

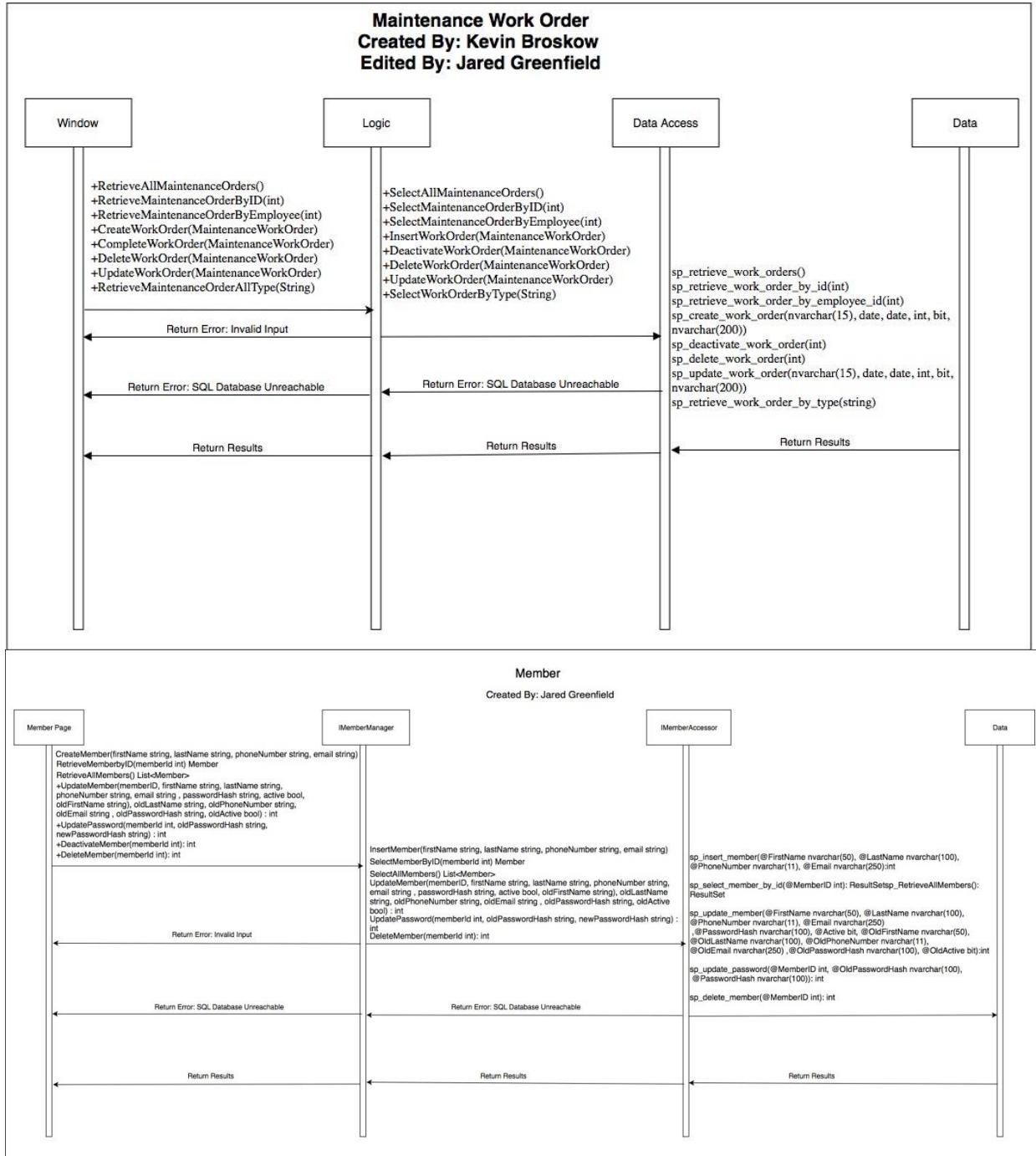


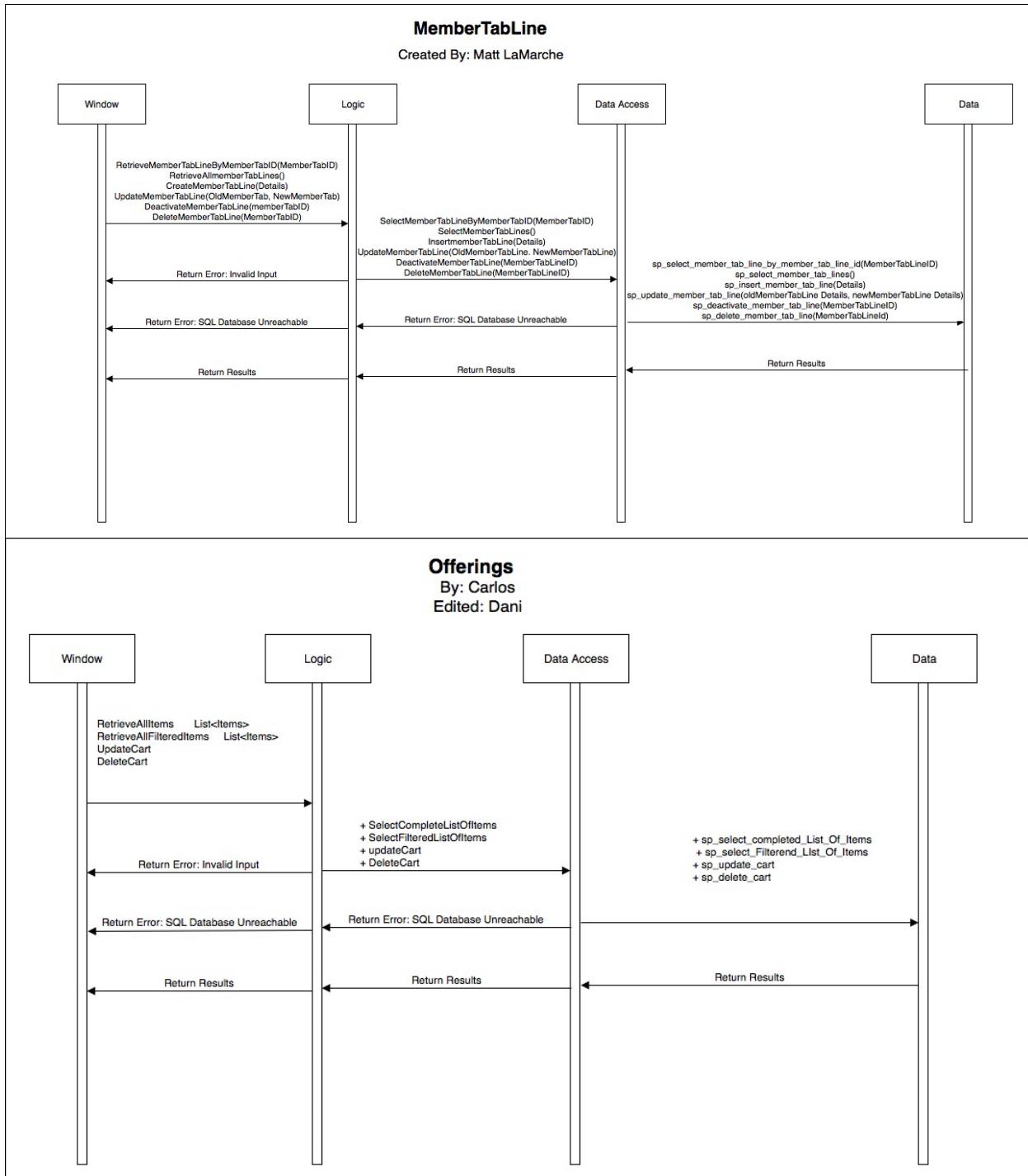
**ItemSupplier**

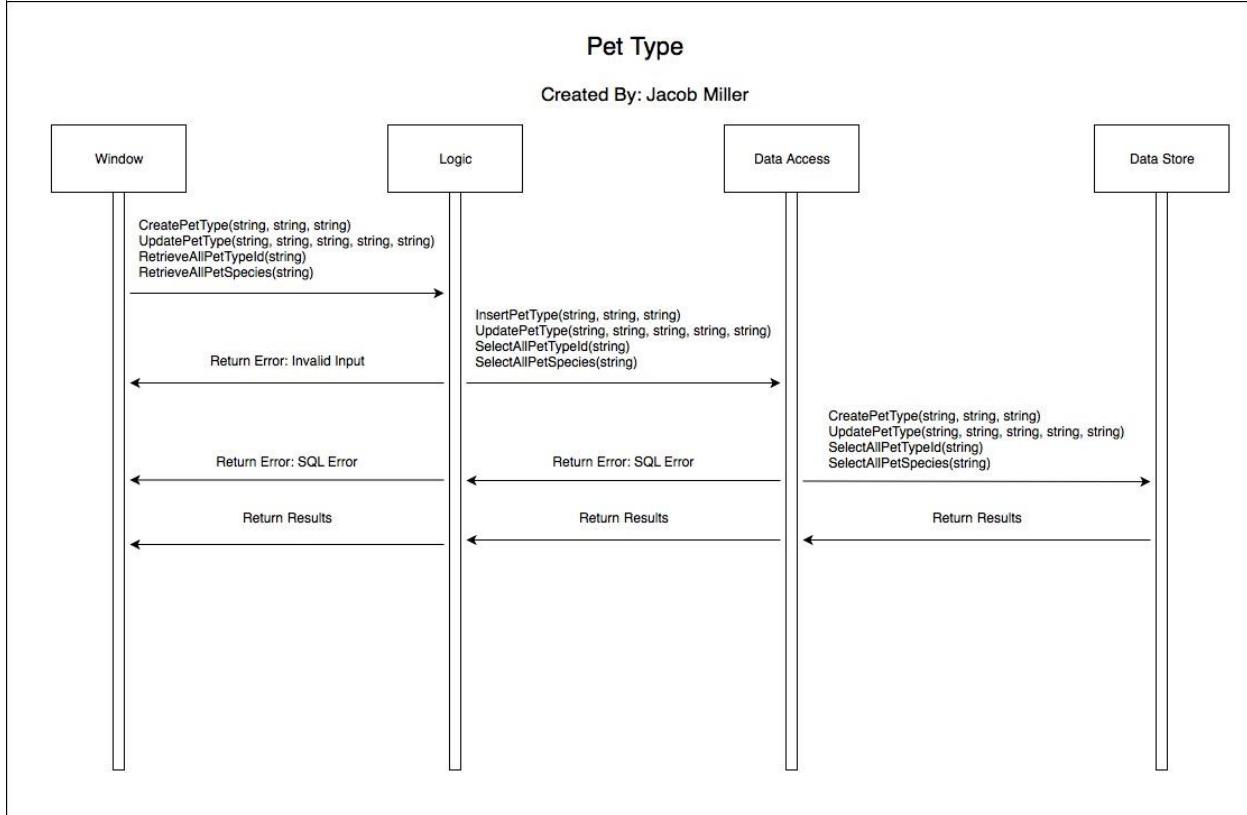
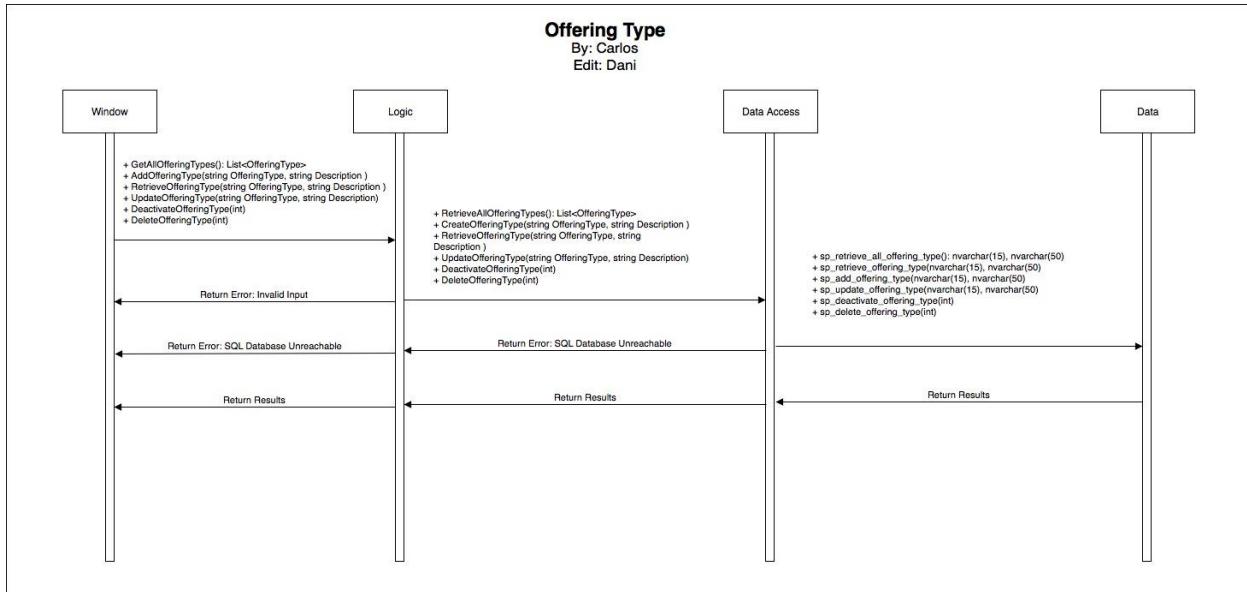
By: Chase

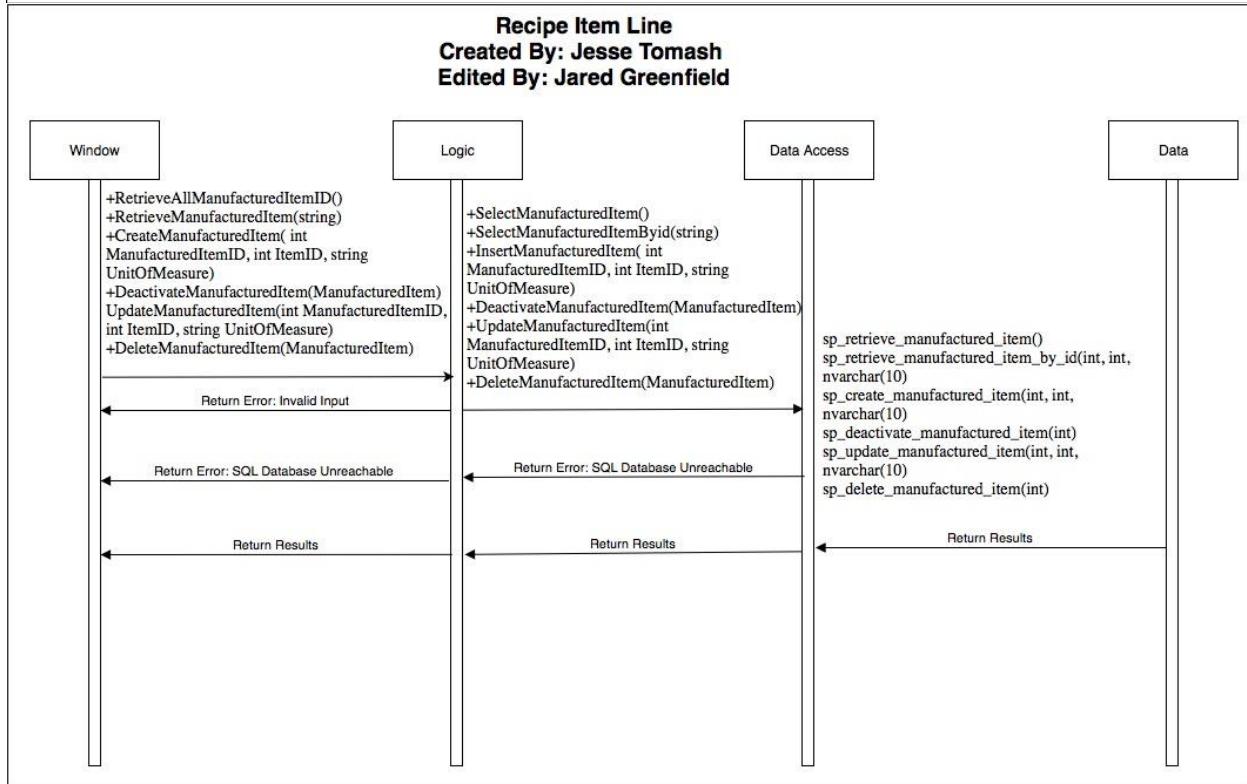
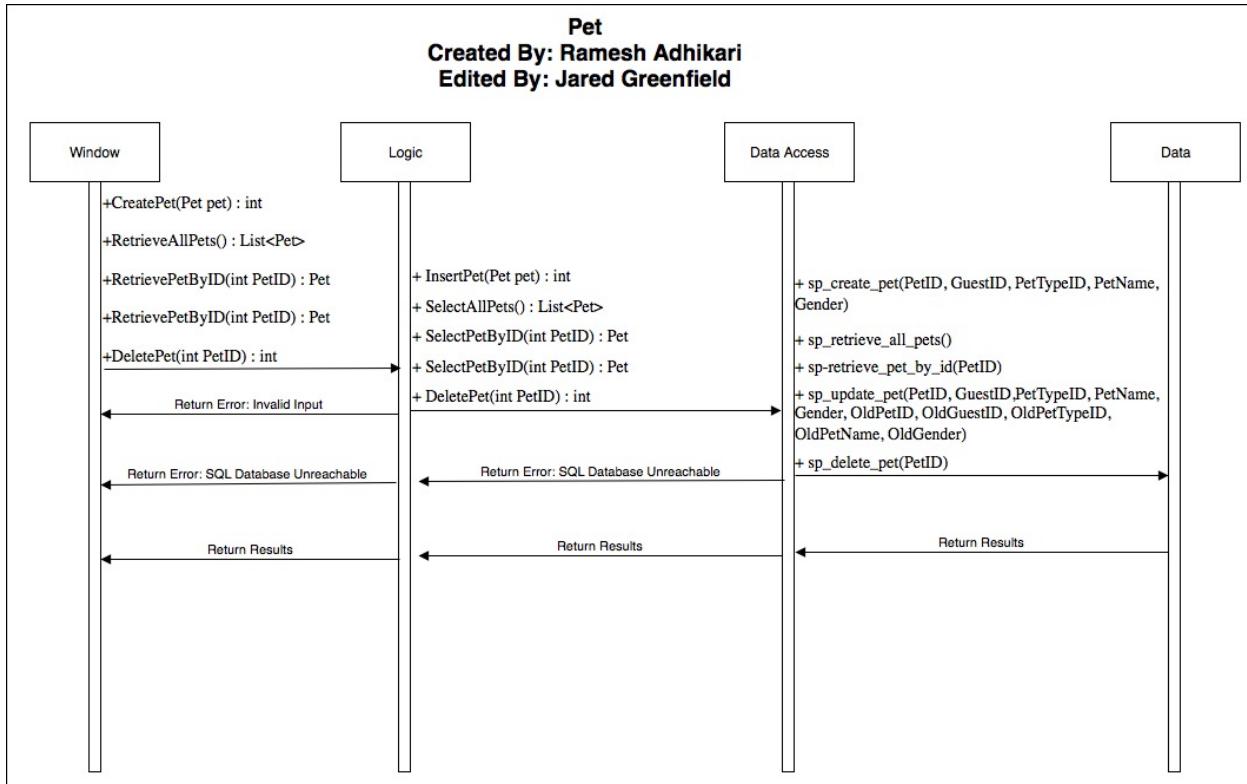


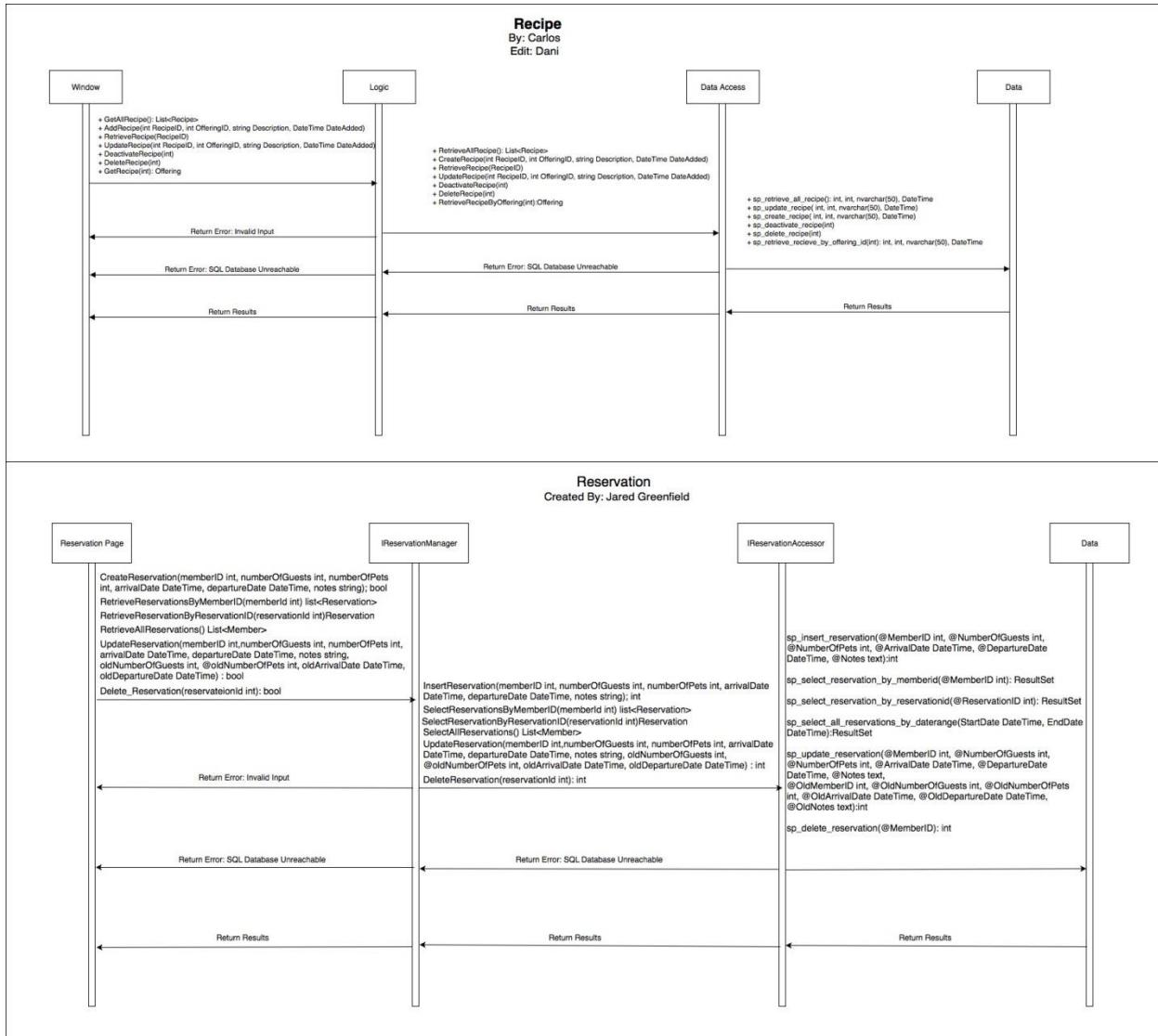




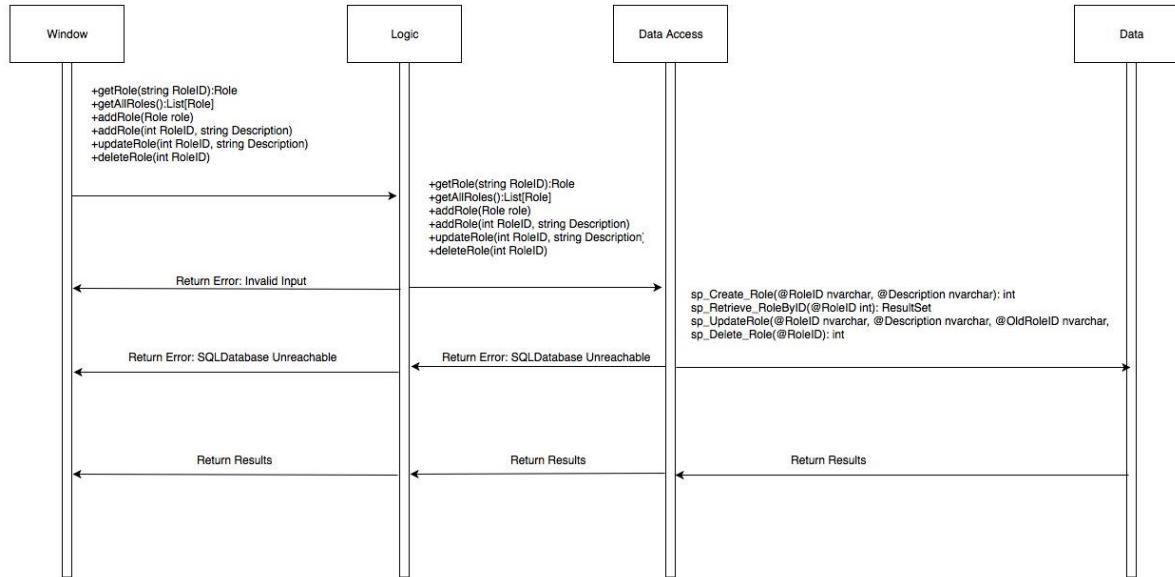




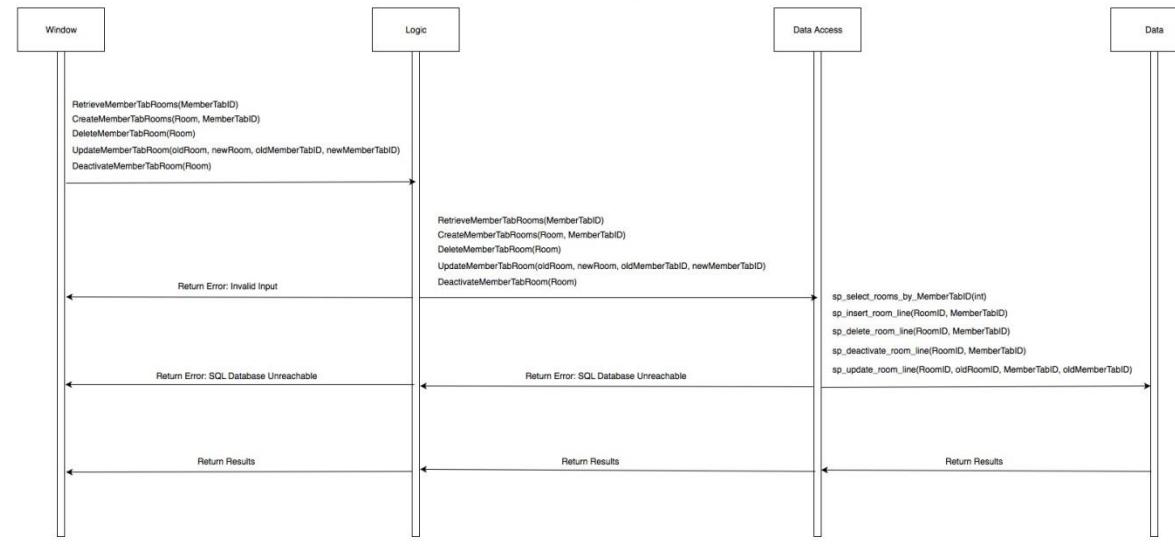


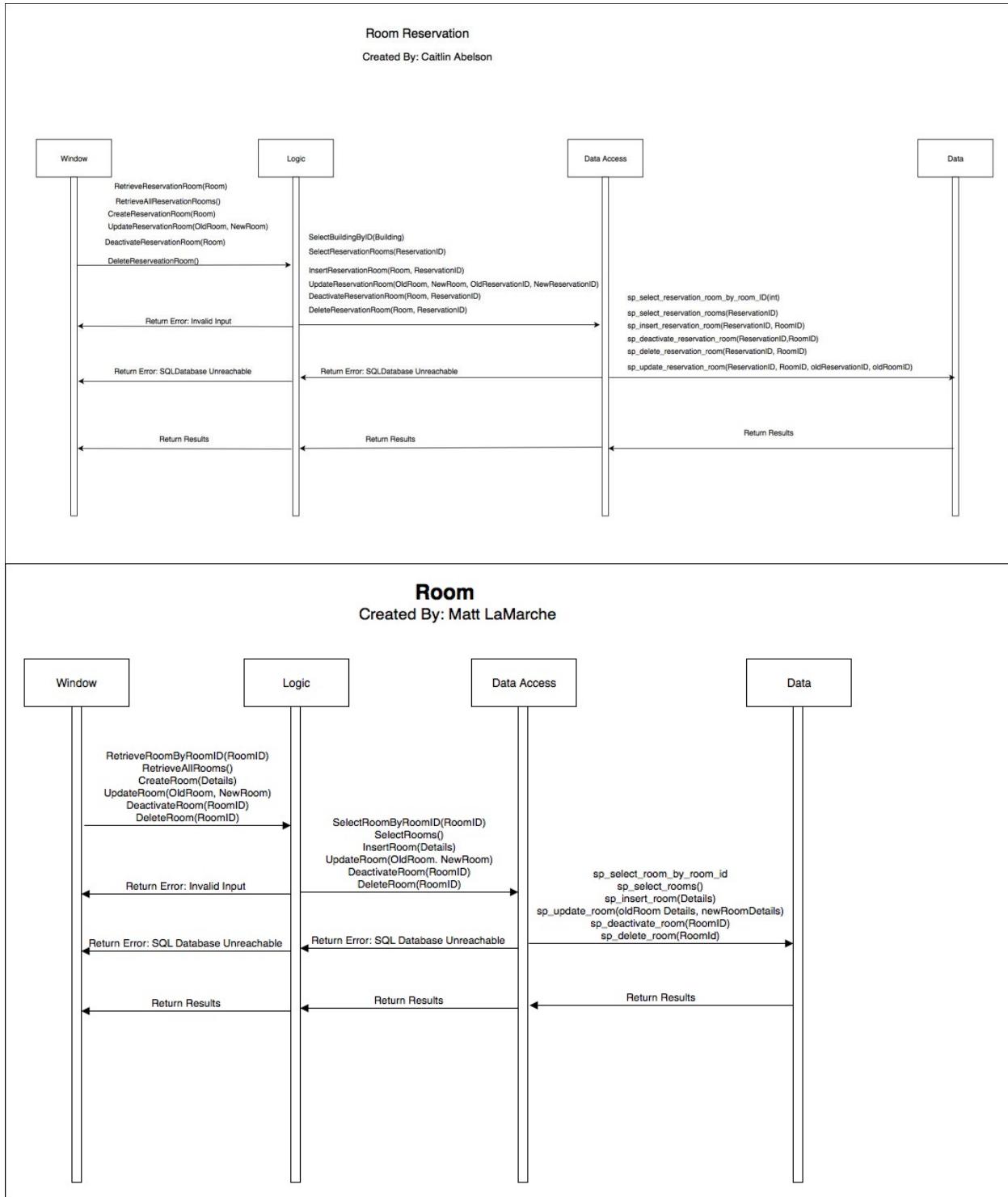


**Millennial Resorts Role Sequence Diagram**  
**Role**  
**Created By: Austin Delaney**

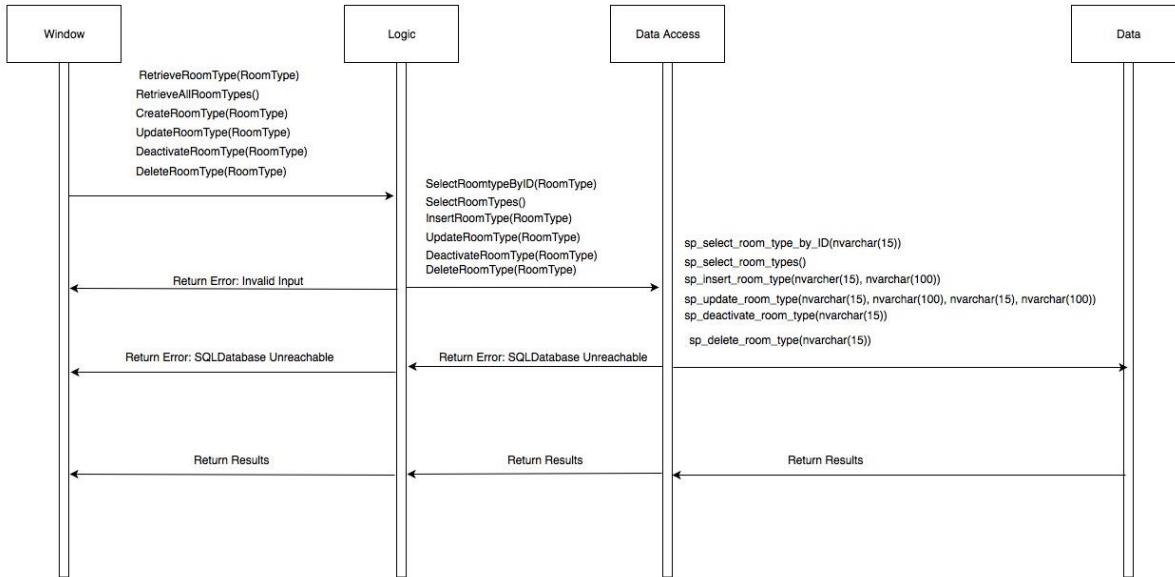


**Room Line**  
**Created By: Caitlin Abelson**



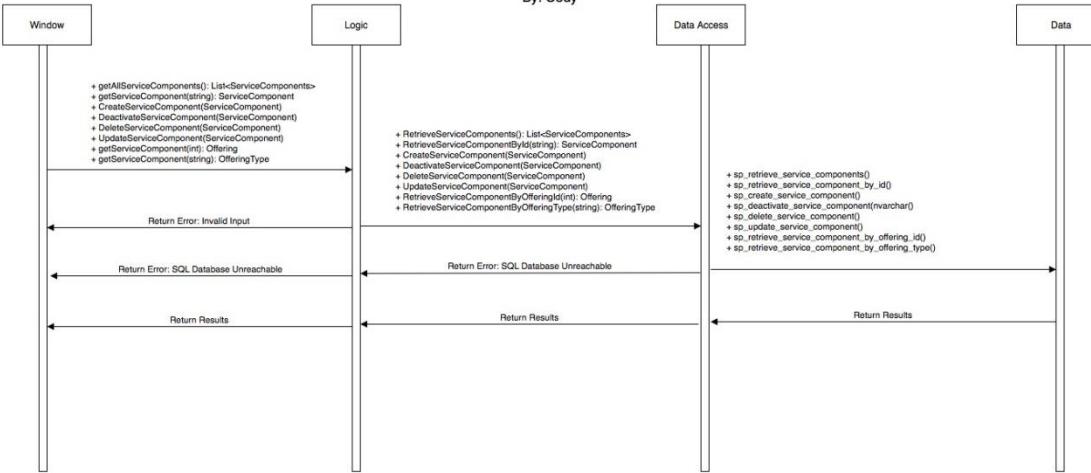


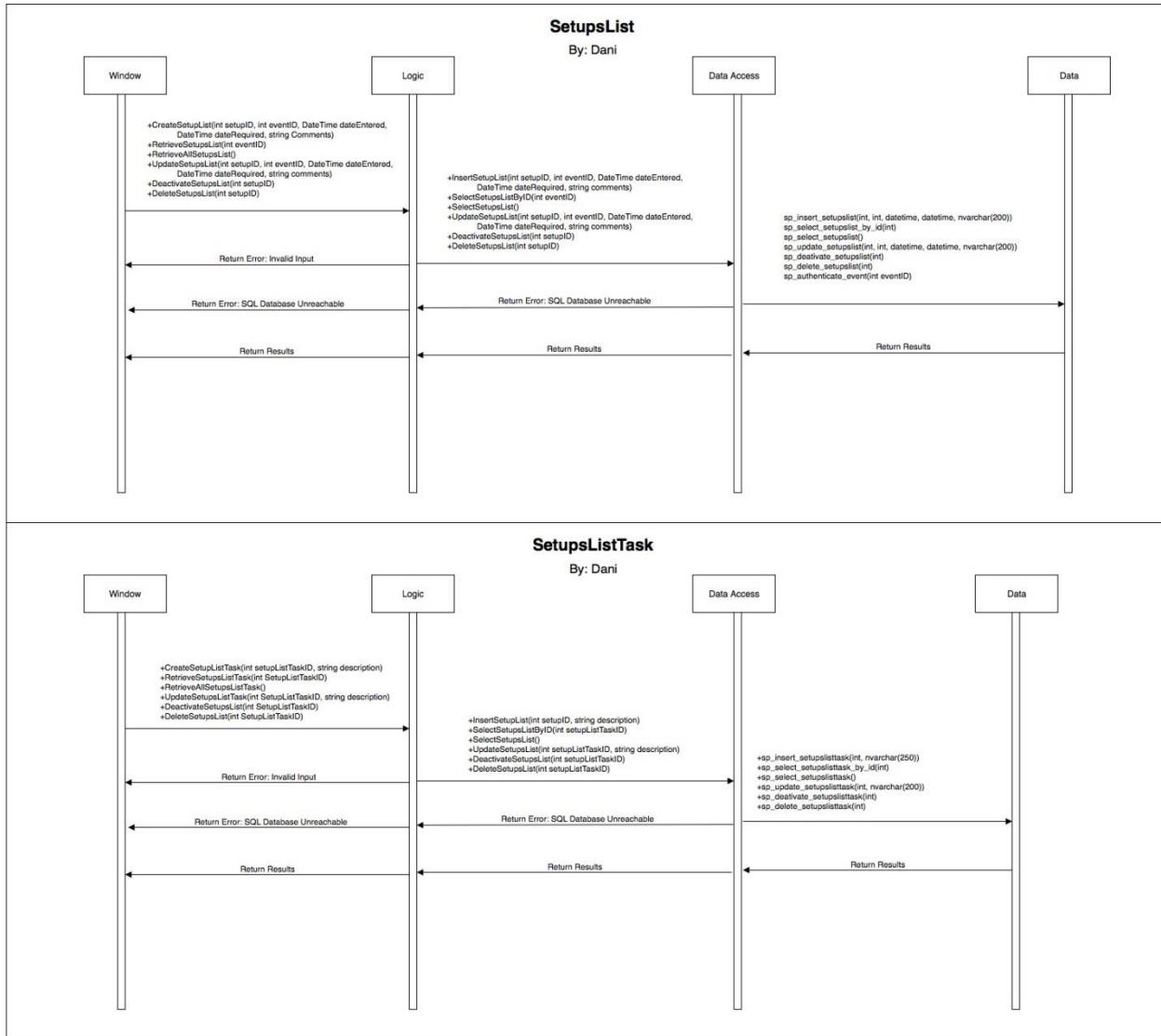
**Room Type**  
Created By: Caitlin Abelson

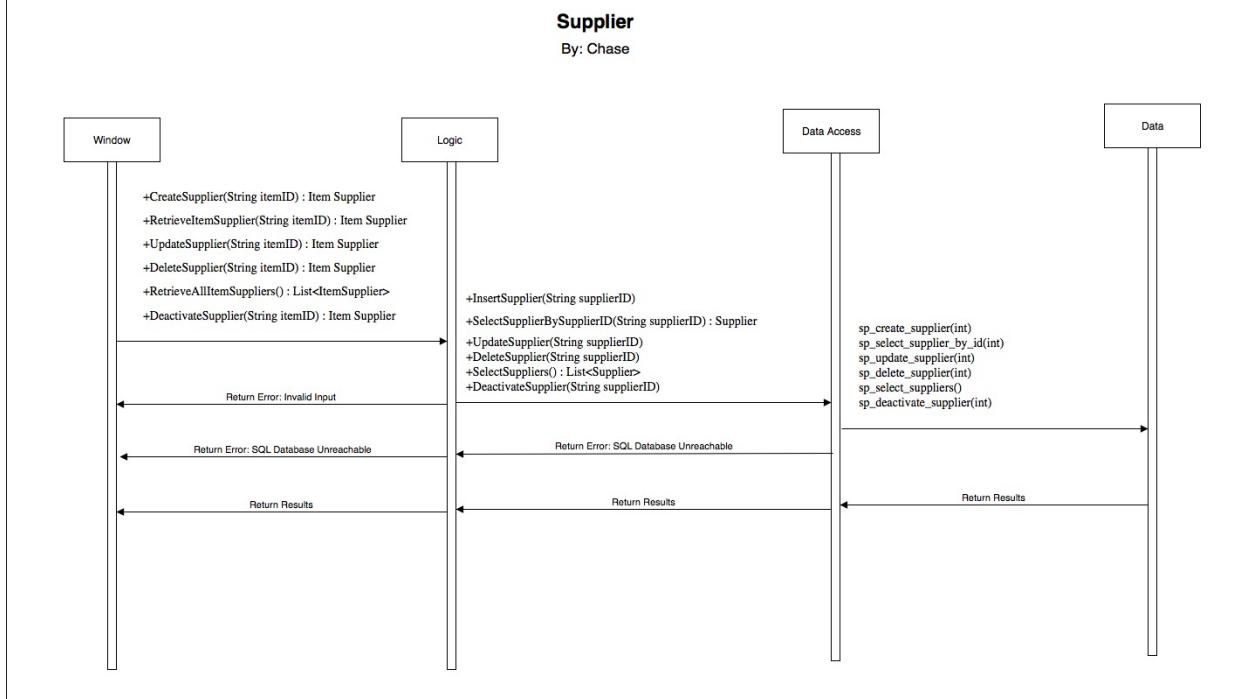
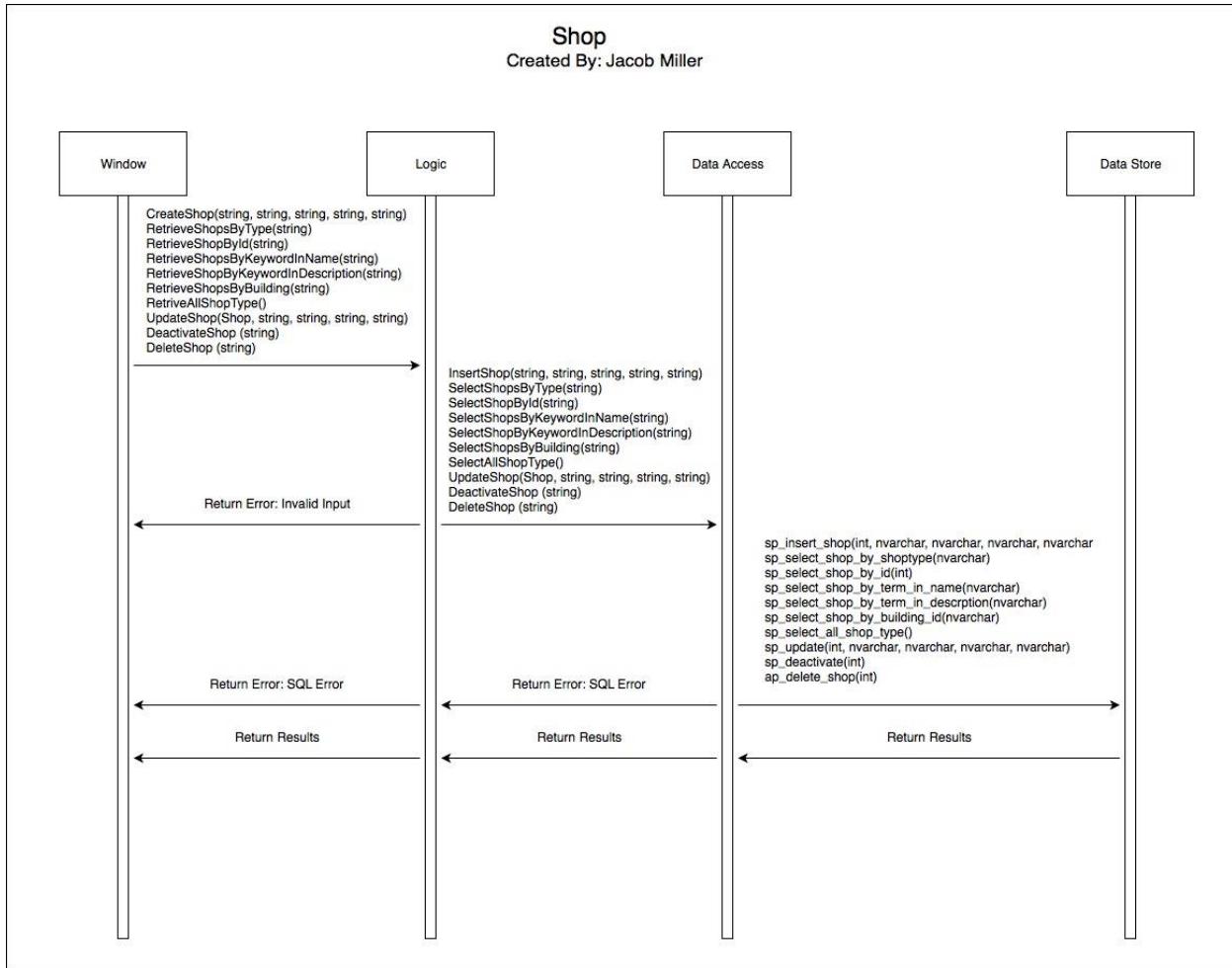


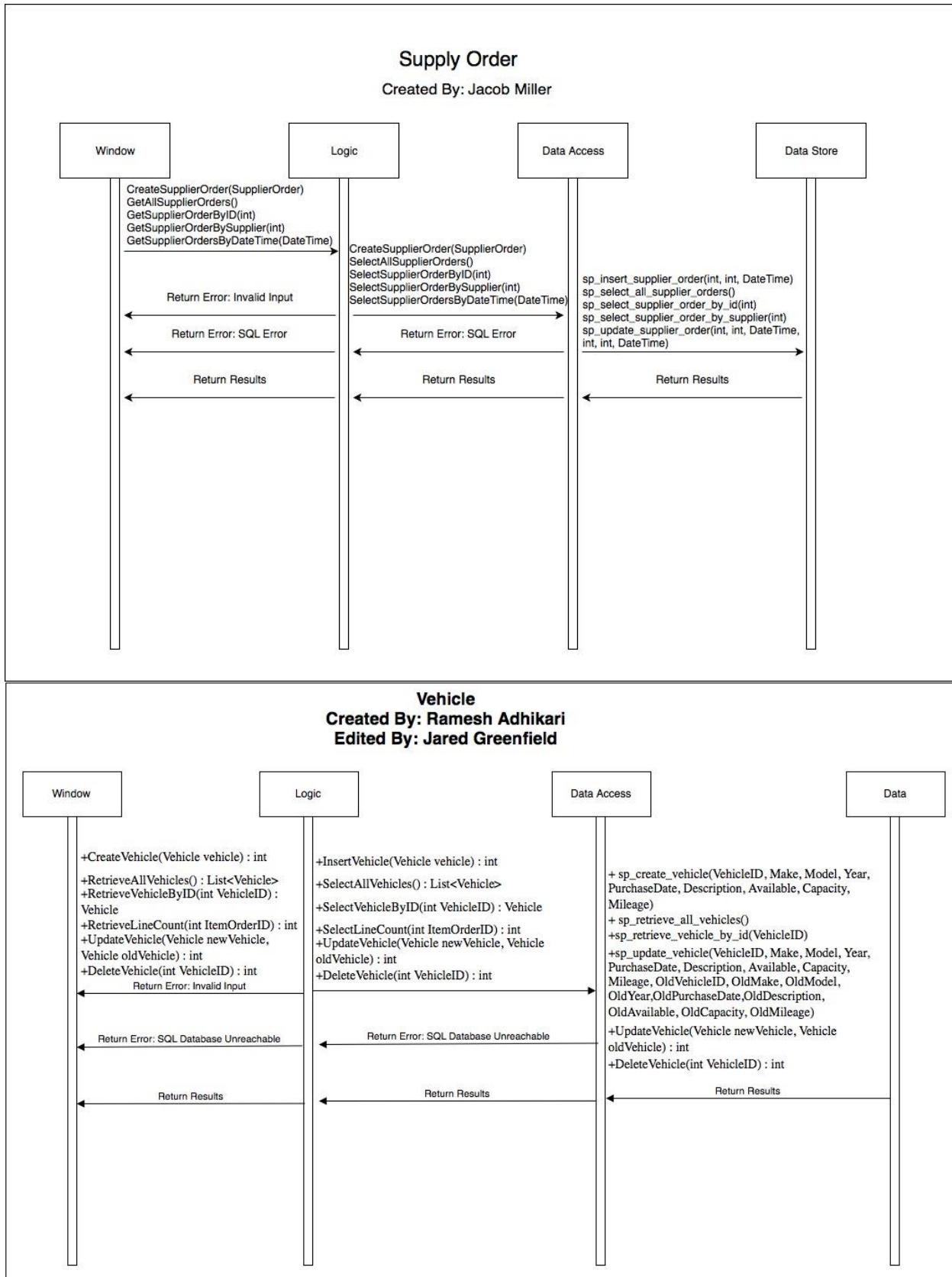
**ServiceComponent**

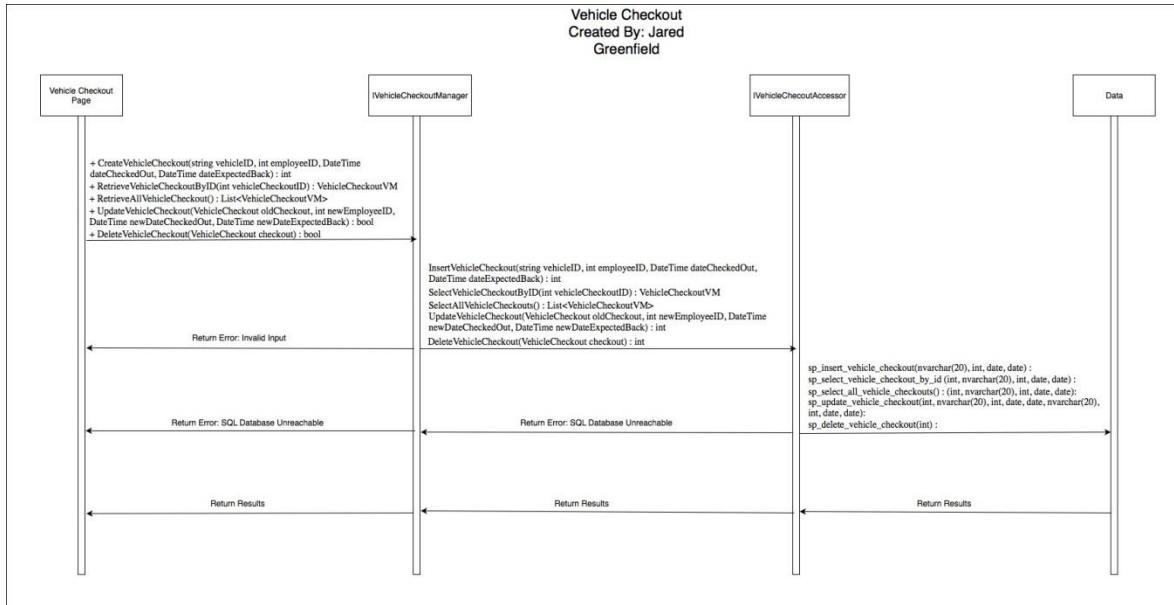
By: Cody









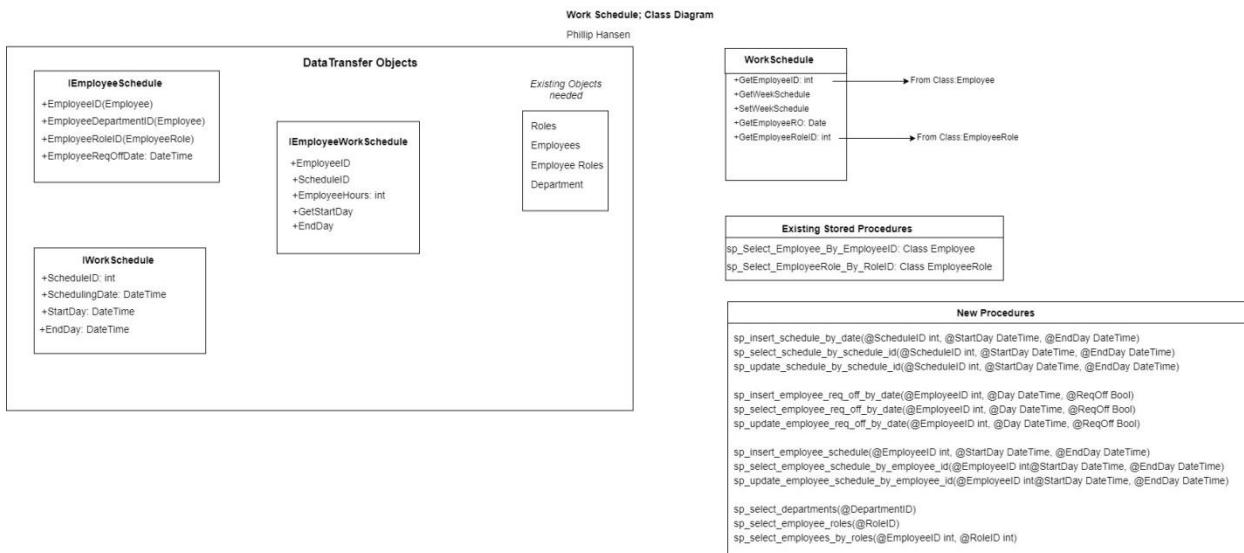


# Automation

## Work Scheduling

This automated feature to create weekly work schedules based upon scheduled entries done by supervisors and the expected need dictated by expected guests at the resort. It will show by department who is scheduled for the week and whether there are too many or too few employees scheduled. This is to make it easy for supervisors to create data driven work schedules.

### Class Diagram

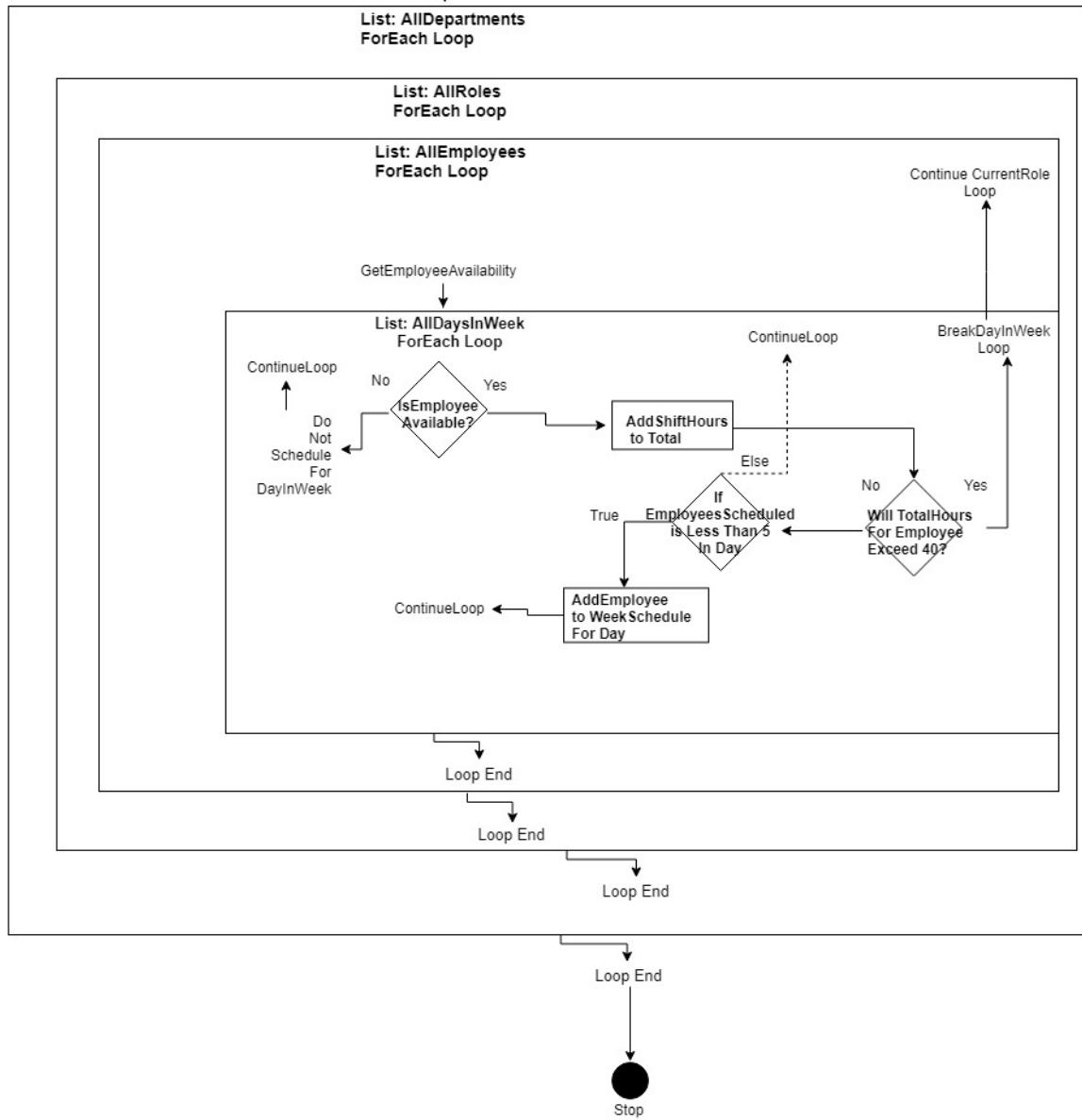


## State Diagram

Work Schedule; State Diagram

Phillip Hansen

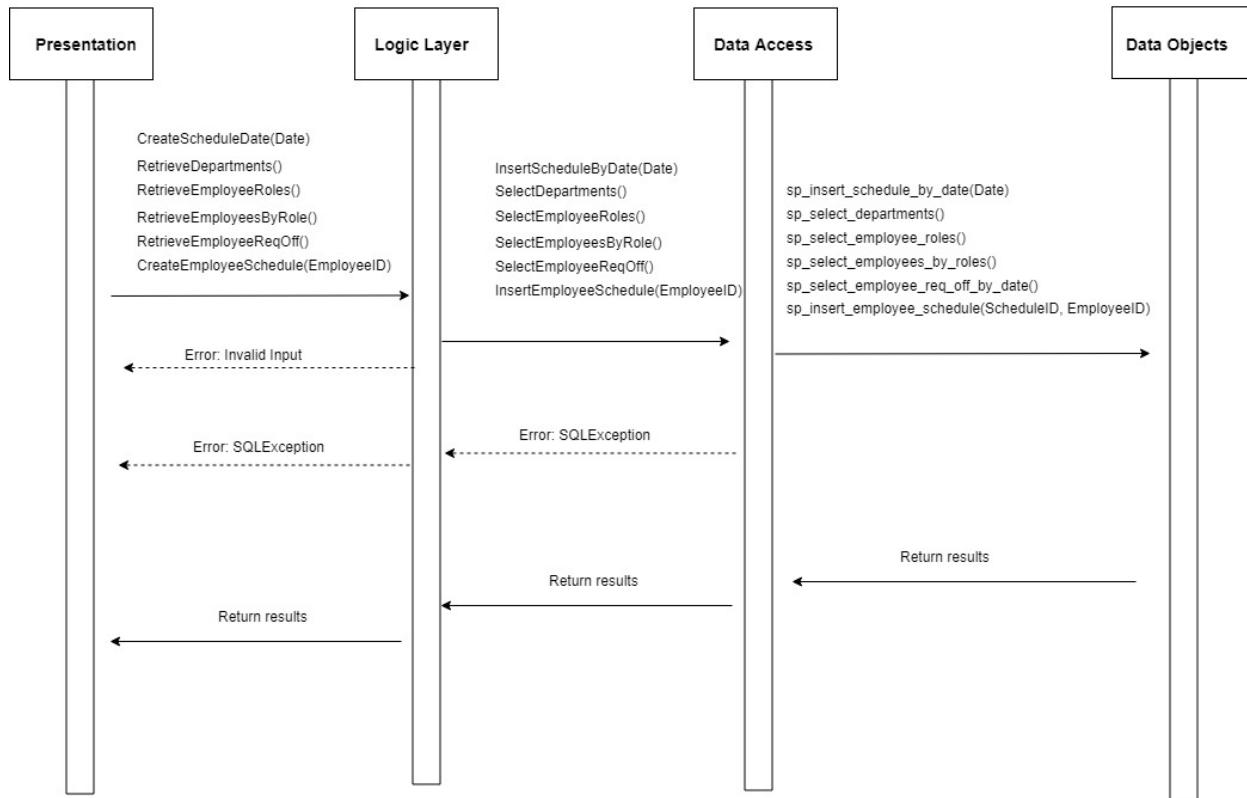
Improved version



## Sequence Diagram

Work Schedule; Sequence Diagram

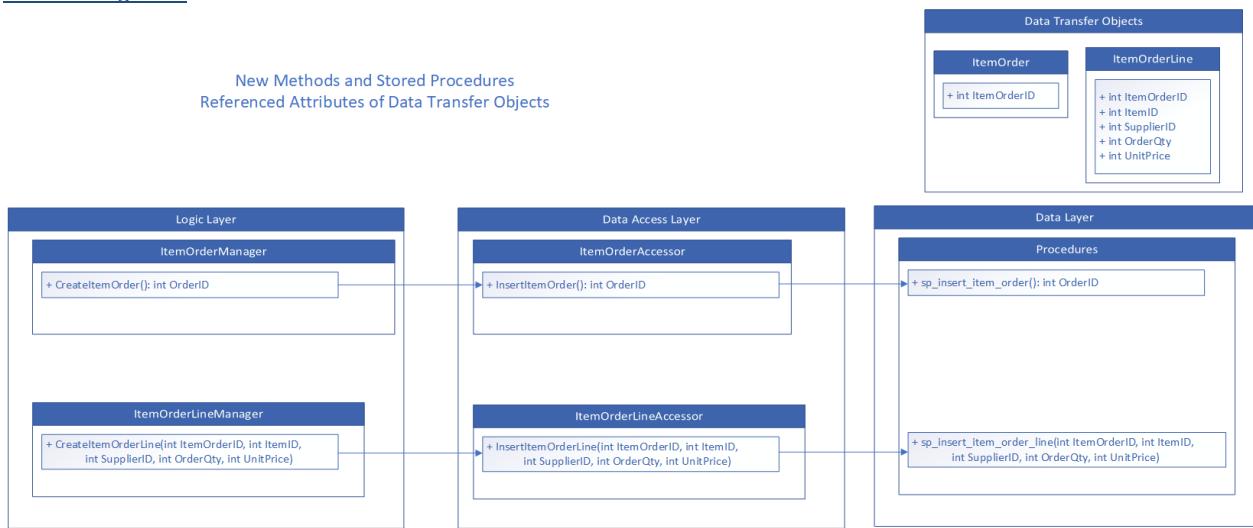
Phillip Hansen



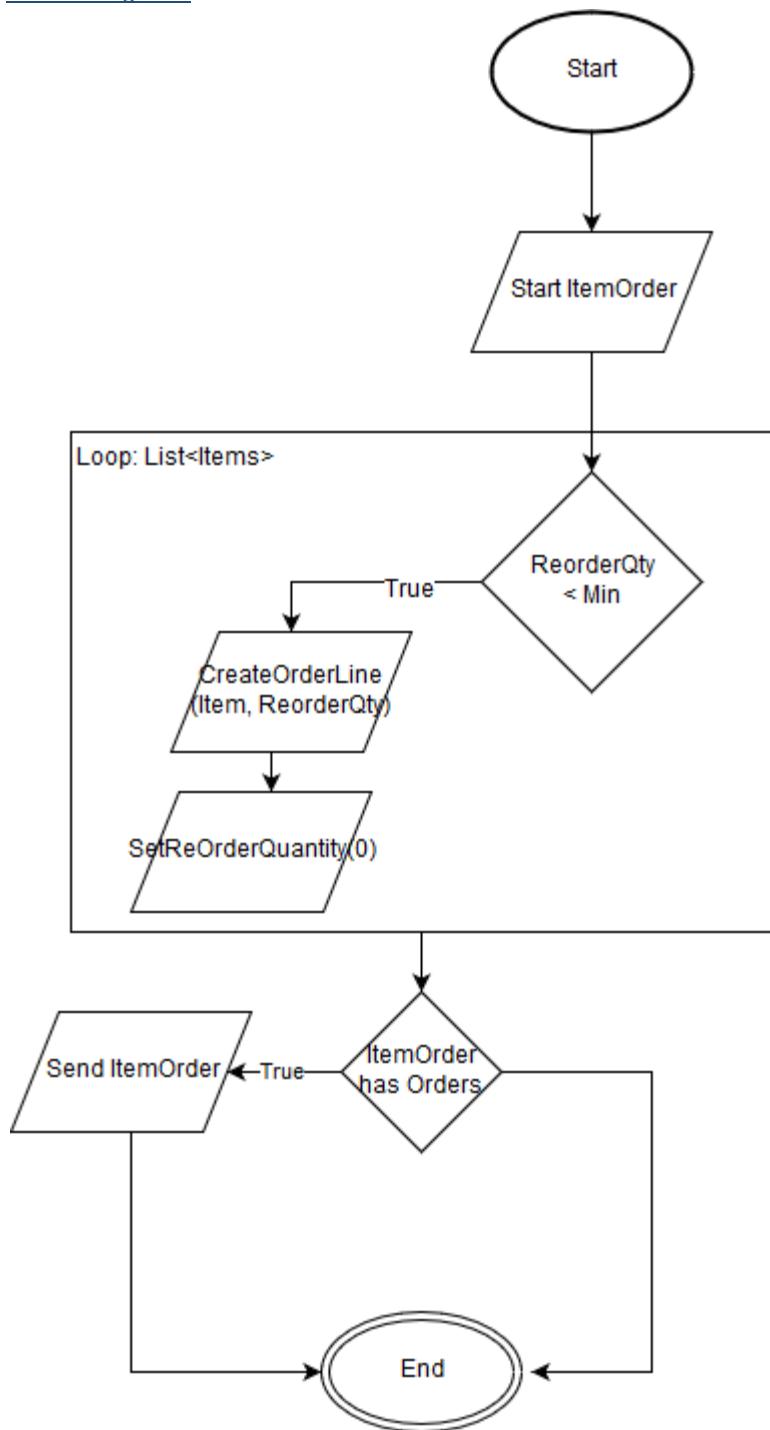
## Ordering Supplies

This feature will automate ordering supplies from vendors. It is based upon hotel reservations and scheduled events and past history at the resort. It will calculate out all the orders and present a list of purchase orders to be reviewed by the purchaser. This will save a considerable amount of time in the ordering process.

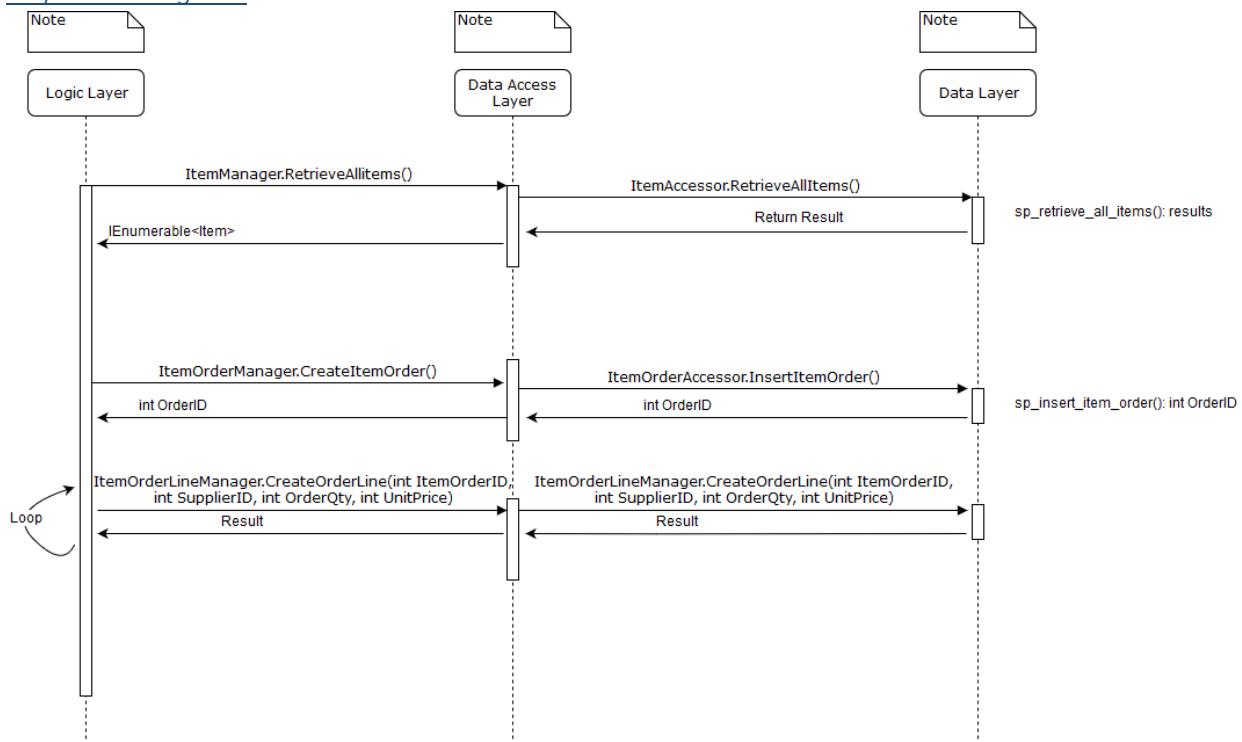
## Class Diagram



State Diagram



## Sequence Diagram

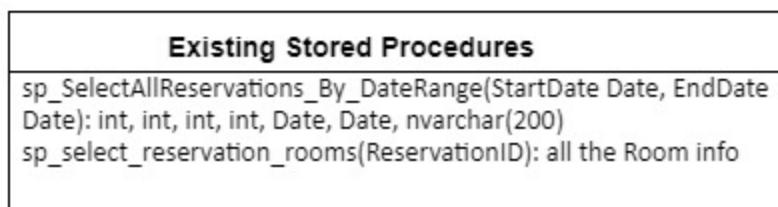
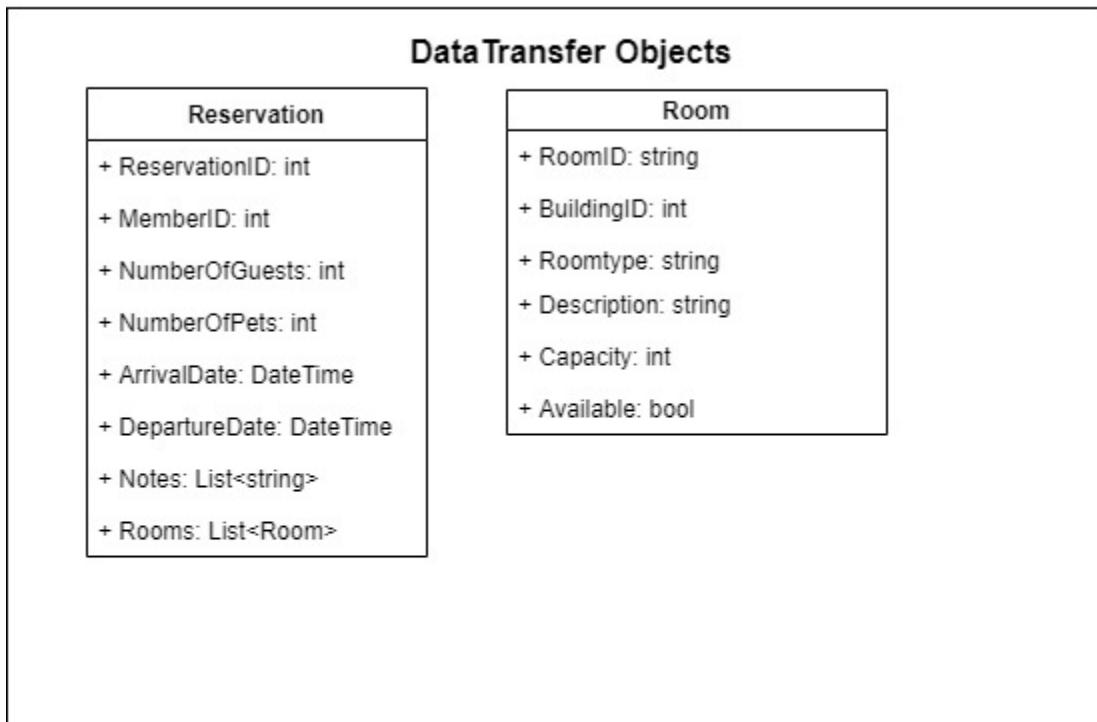


## Reservations

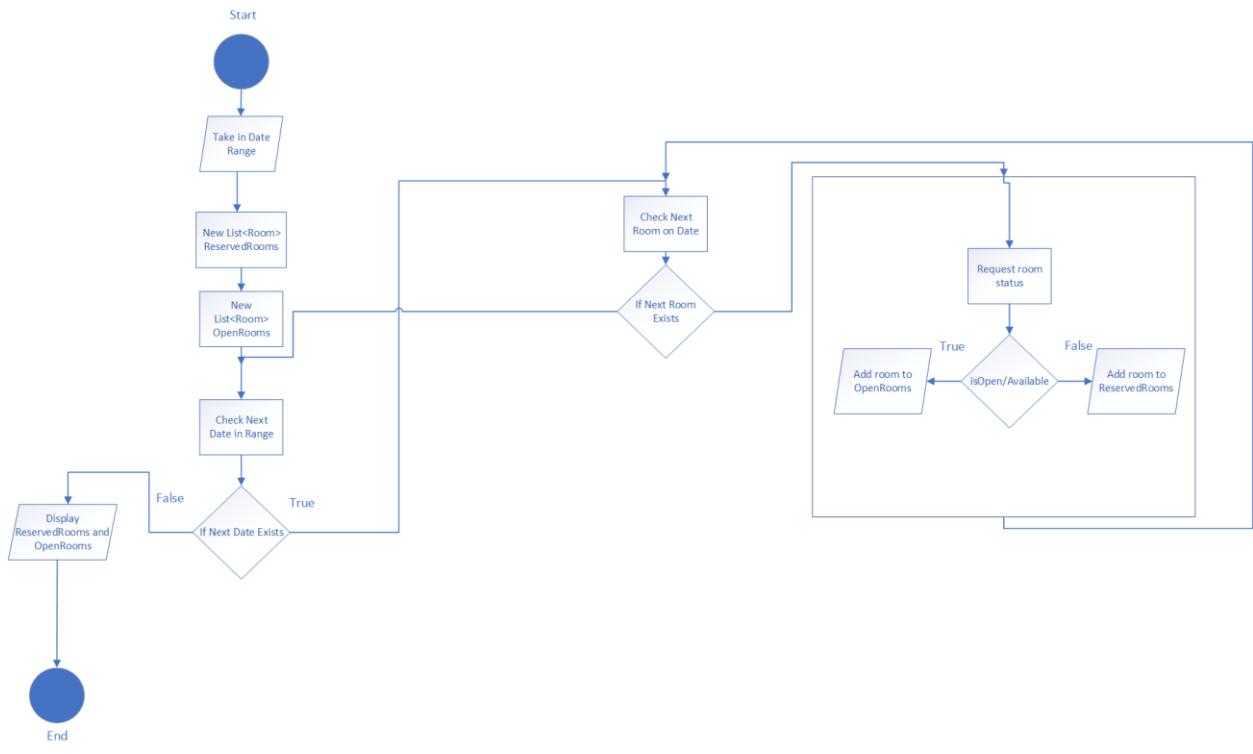
The automation for setting up reservations on a timed schedule. This feature will allow for ease of making reservations by providing an overview of available rooms based upon date ranges.

## Class Diagram

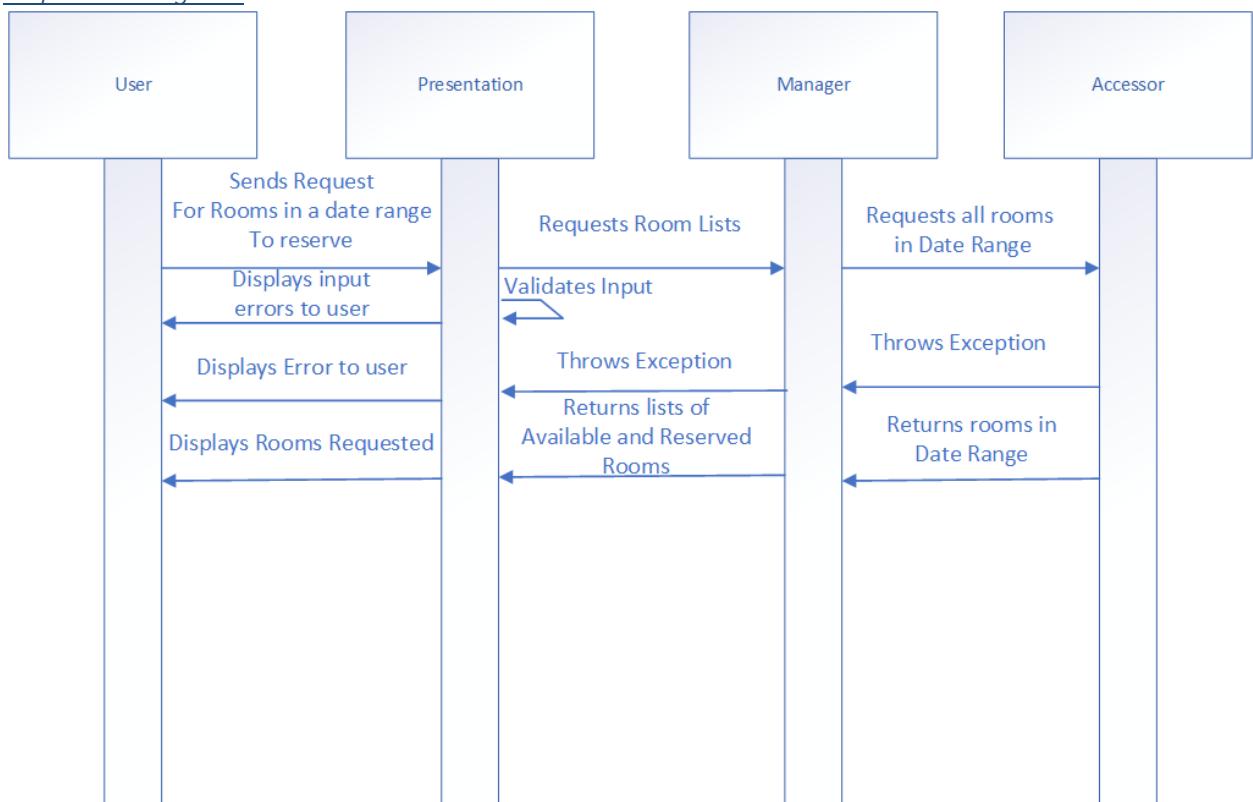
### Reservation Class Diagram



### State Diagram



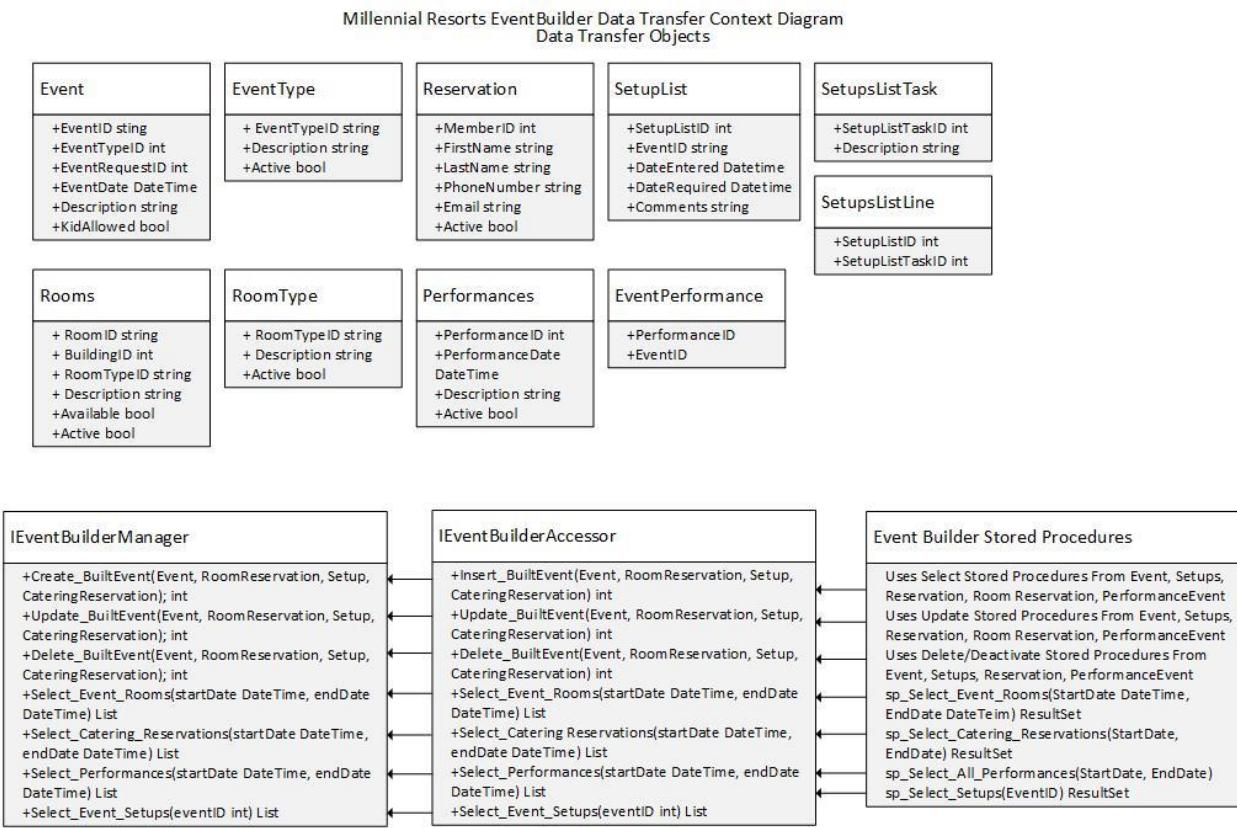
### Sequence Diagram



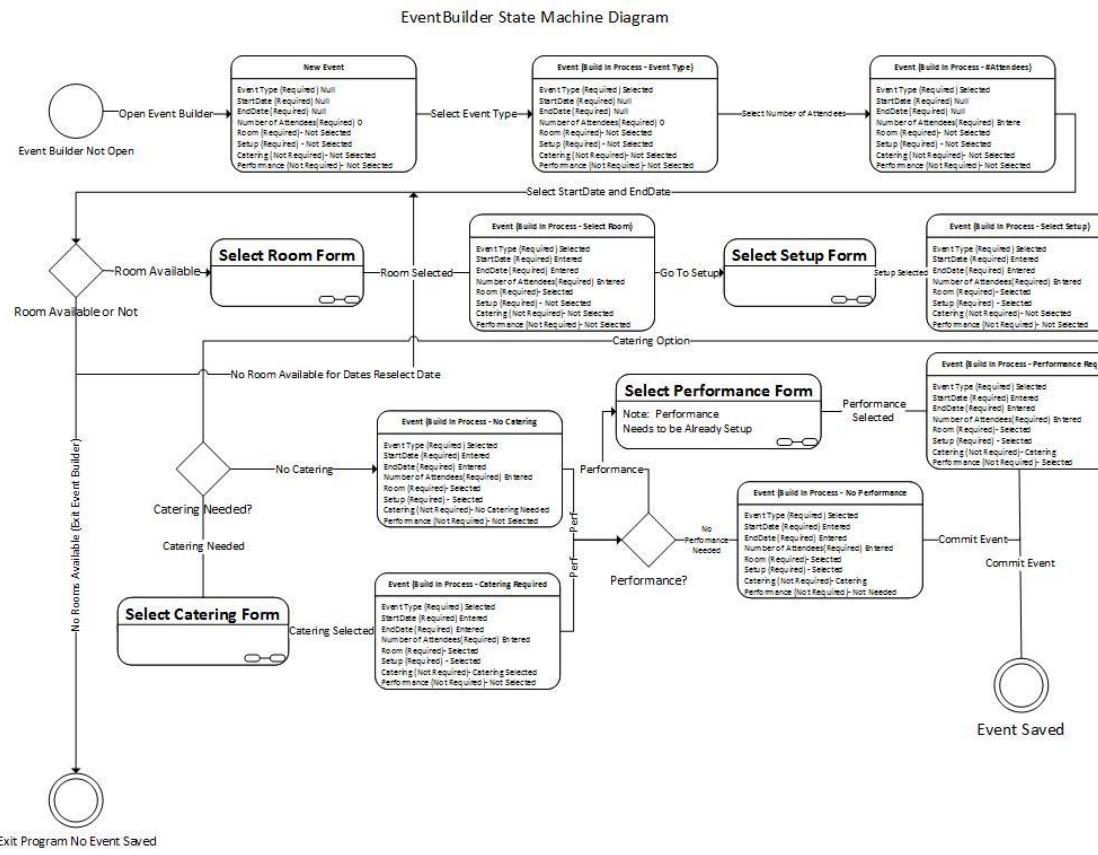
## Event Builder

The Event Builder is way for an event planner to build all the necessary elements of an event in one location. It will walk the planner through scheduling rooms for the event, catering, event setups and adding an event performance. It will let the planner know at each step of the process if the resources are available to accommodate the event. This will streamline event planning.

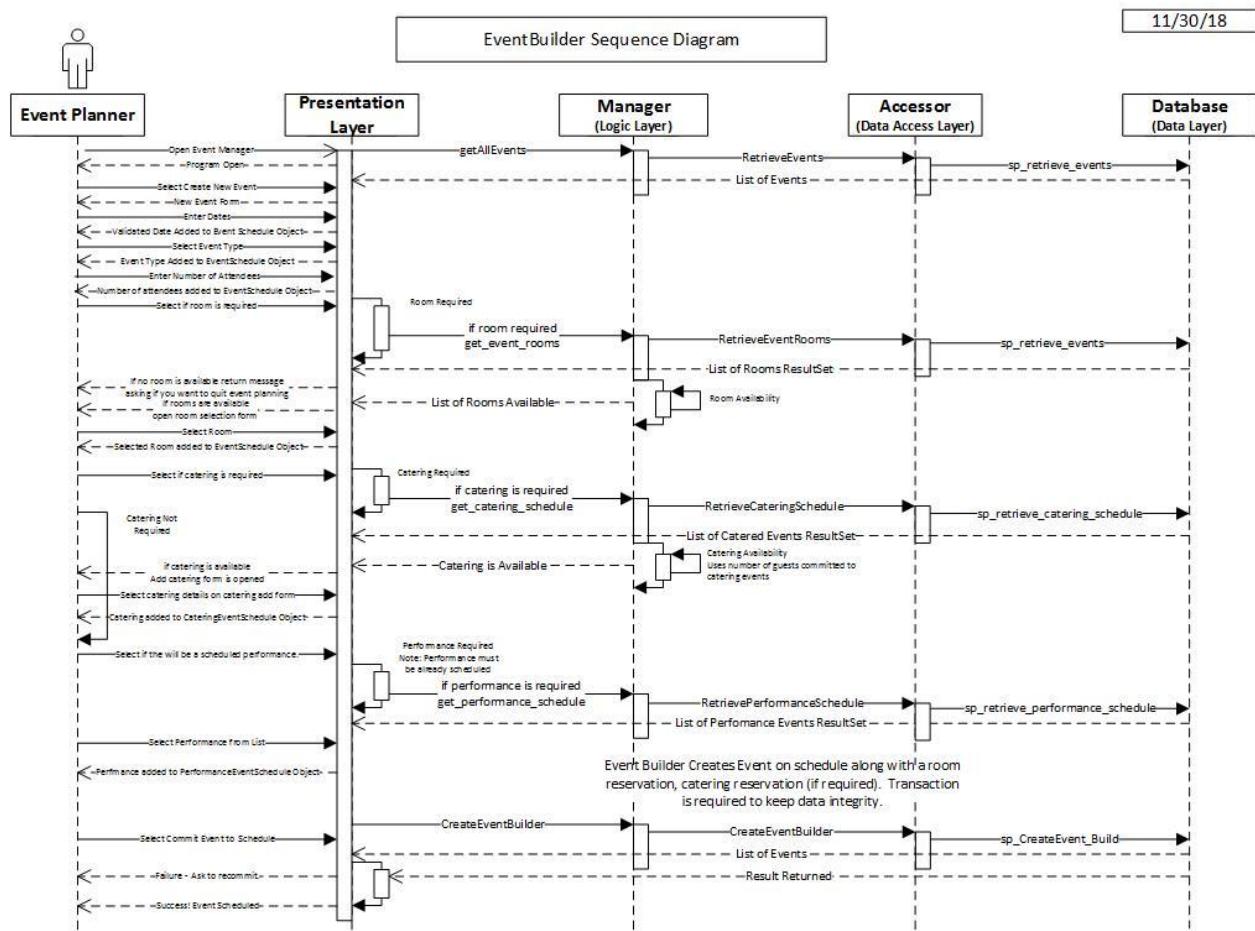
### Class Diagram



## State Diagram

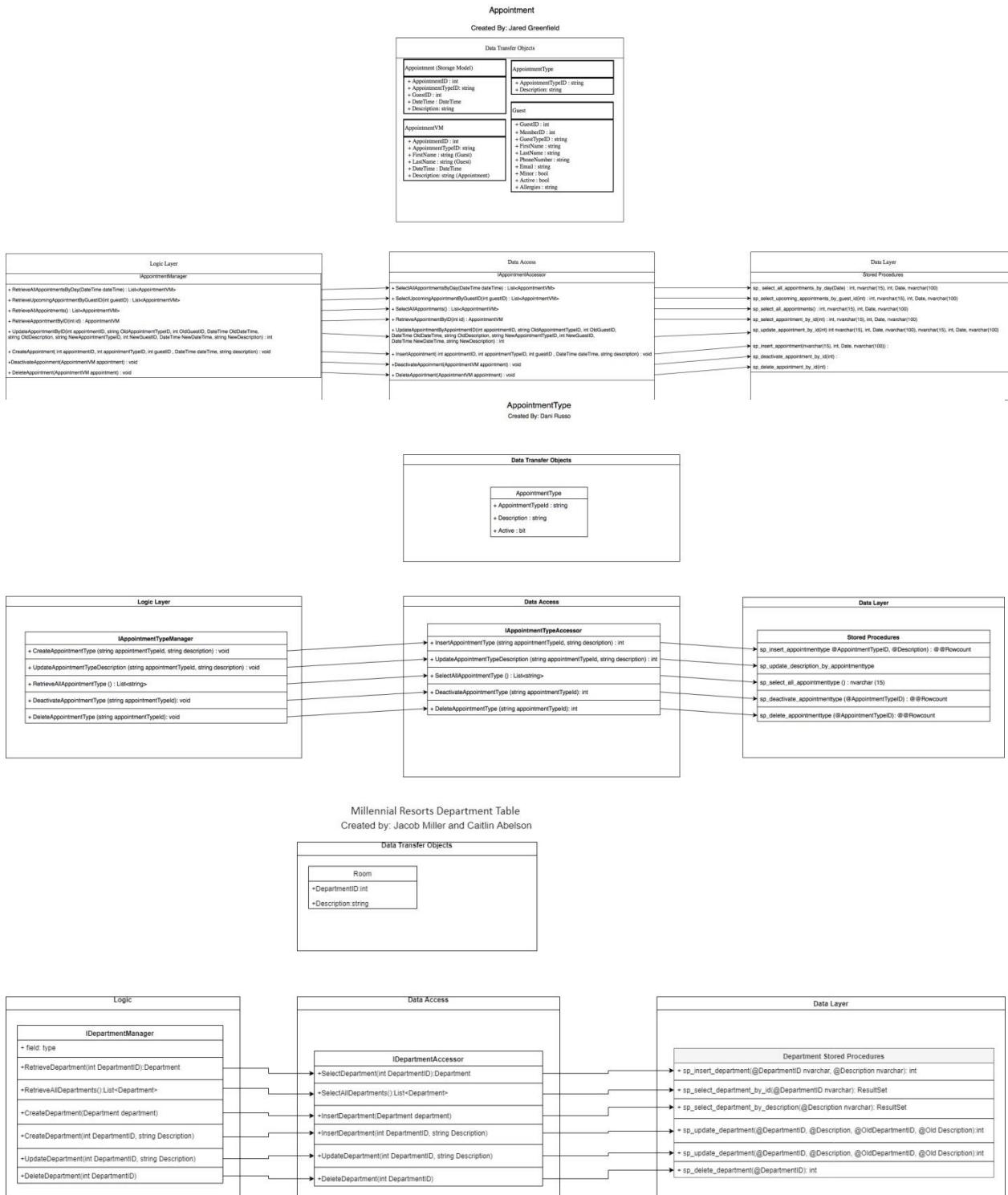


## Sequence Diagram



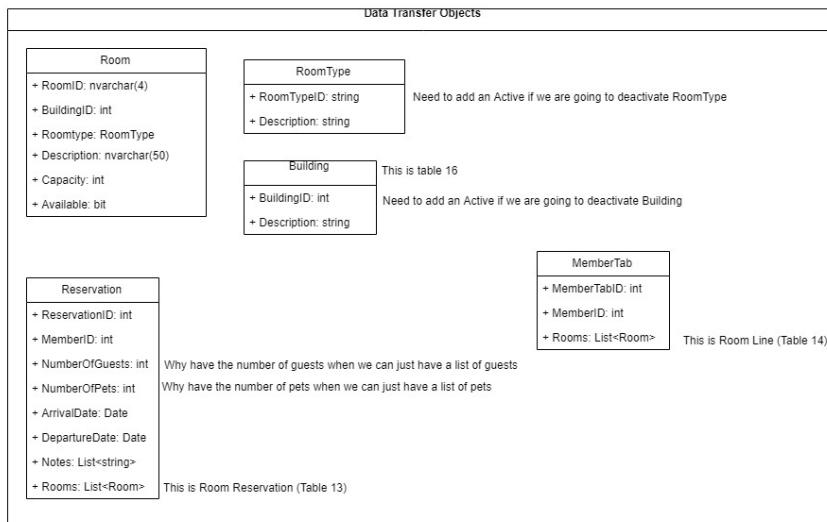
## Class Diagram

The following class diagrams provide a detailed description of the program's components by indicating where data and functionality reside. The operations listed in each class describe the interactions among the entities and illustrate how data is transferred from the database to the user interface. From the diagrams, we can pinpoint similar behaviors improving consistency and efficiency during coding.



## Building

Created By: Matt LaMarche



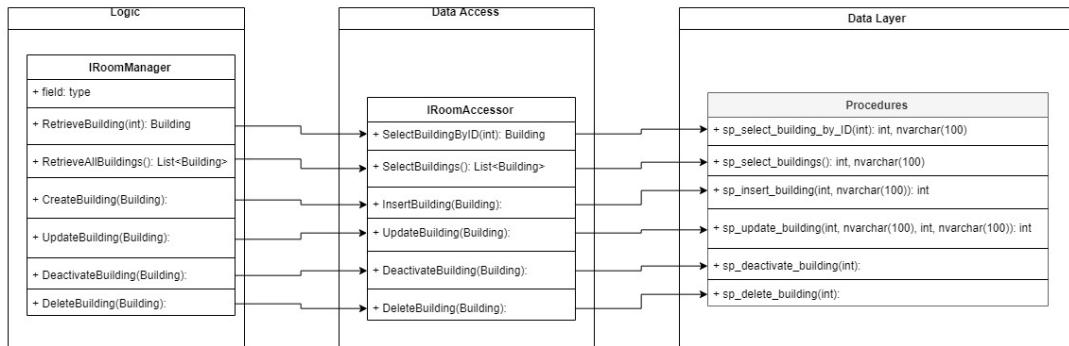
## Building

A Building is an Object created out of the Building Table. The Building is primarily used in the Room Table at the moment so I have included its Methods into the RoomManager Methods in the Logic Layer and RoomAccessor Methods in the Data Access Layer

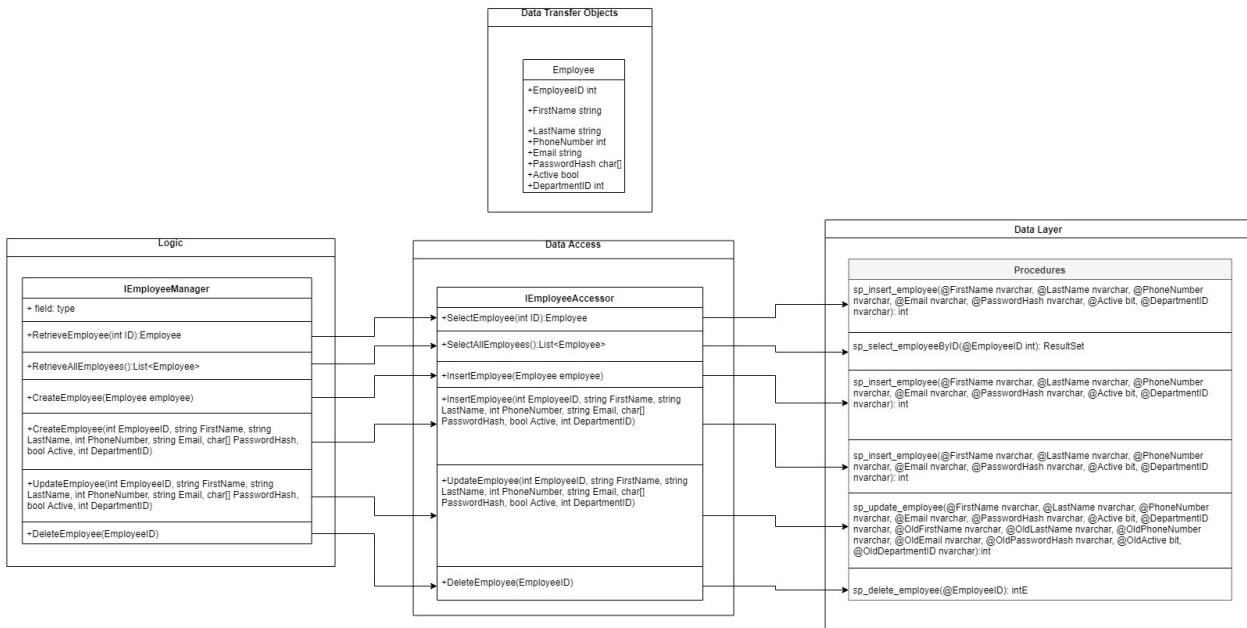
NOTE: Should Building be used in other areas of our program we can pull it out of our RoomManager and RoomAccessor classes to create BuildingManager and BuildingAccessor classes

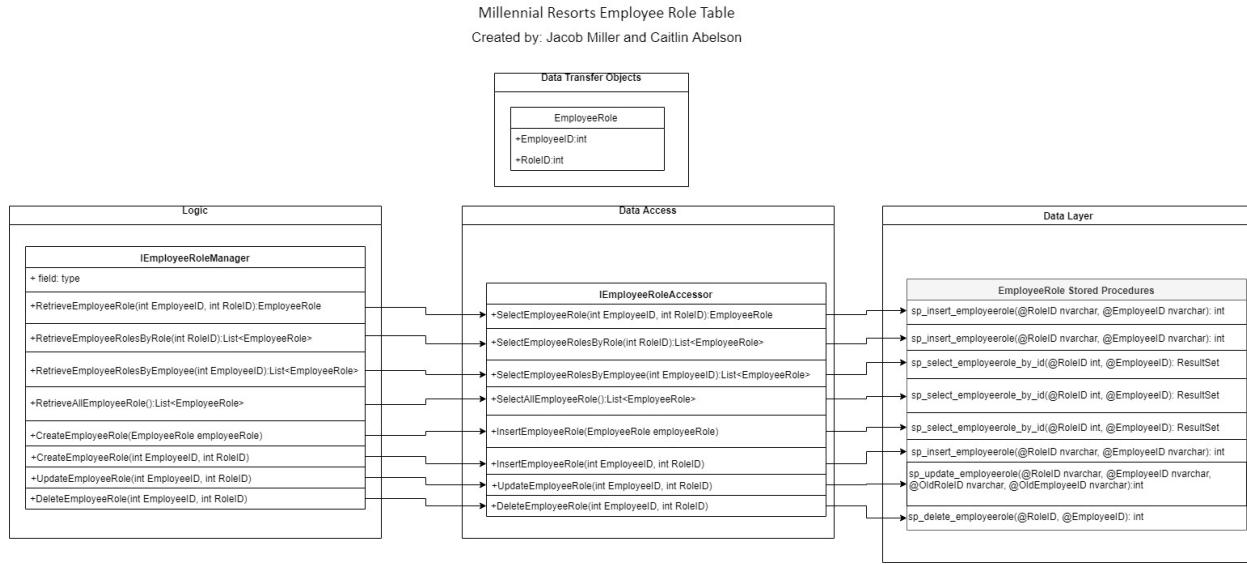
## Tables

Room Reservation  
Room Line  
RoomType  
Building



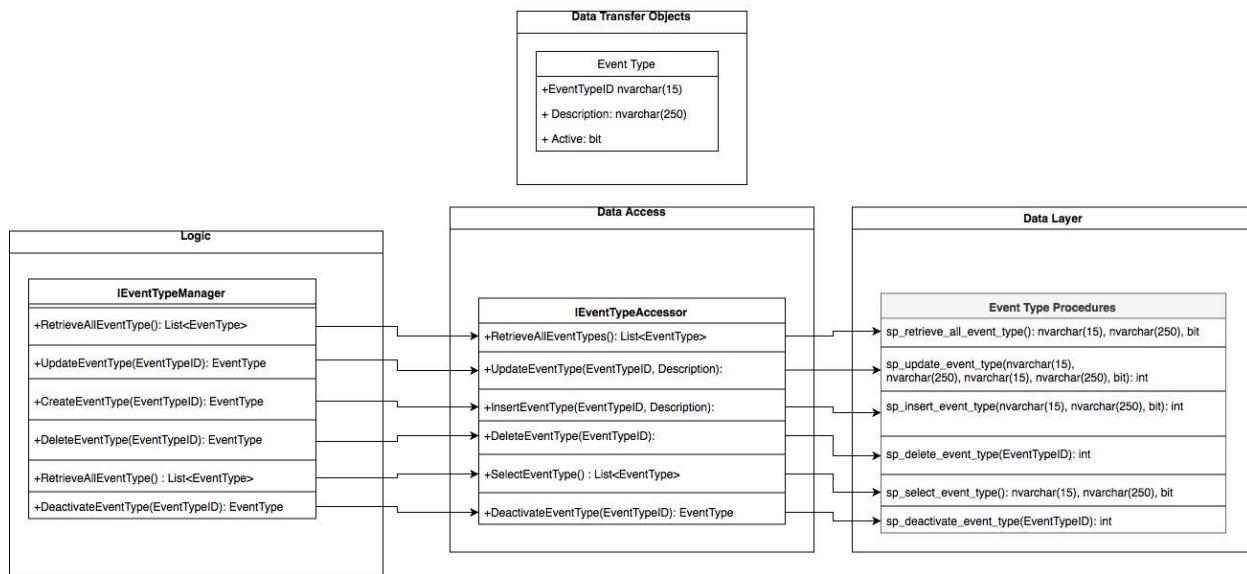
Millennial Resorts Employee Table  
Created by: Jacob Miller and Caitlin Abelson





### Event Type

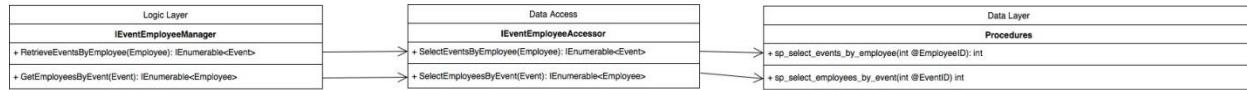
Created By: Kevin B. and Caitlin Abelson



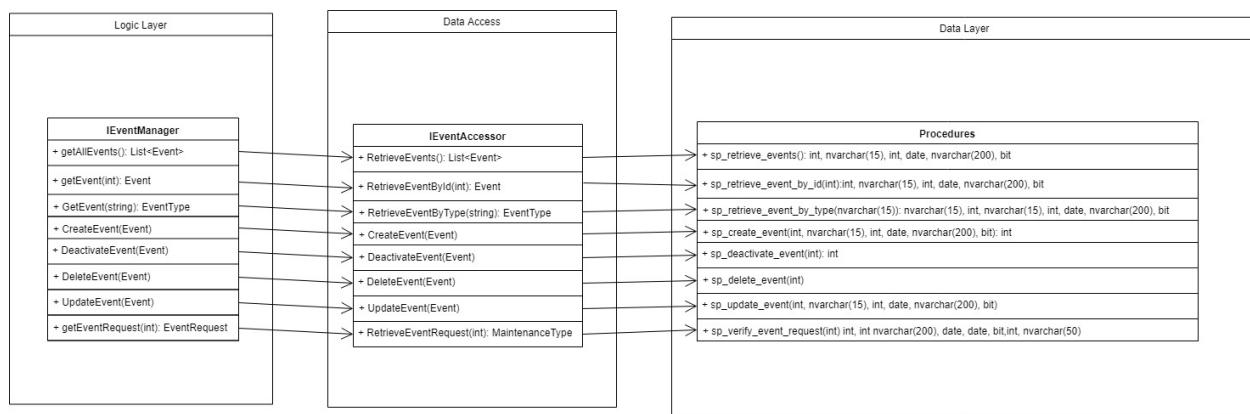
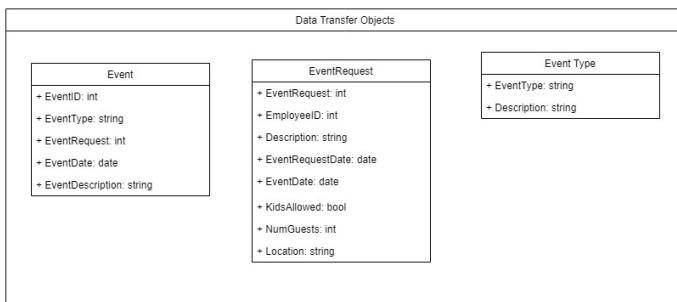
## Event Employee

Created by: James Heim

THIS IS A JOIN TABLE

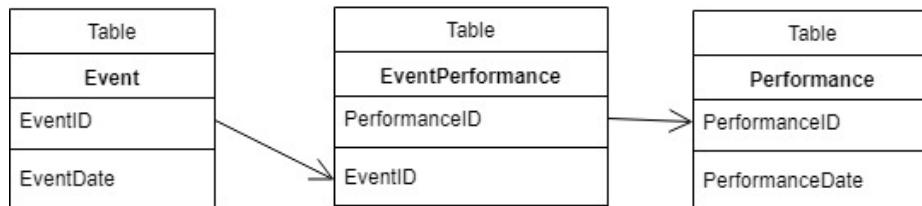


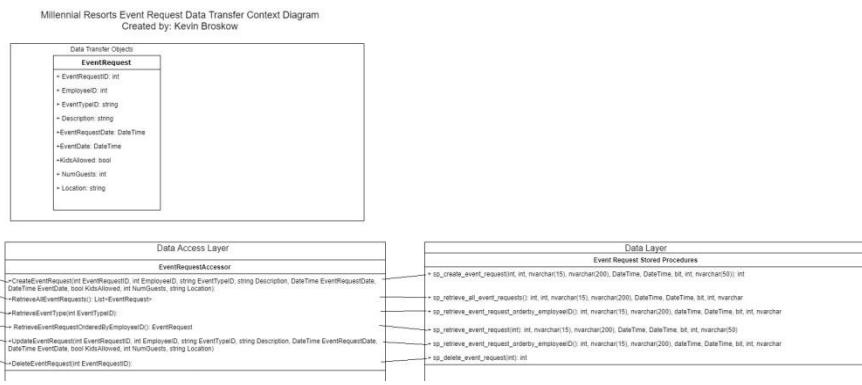
**Event**  
Created By: Caitlin Abelson



## Event Performance

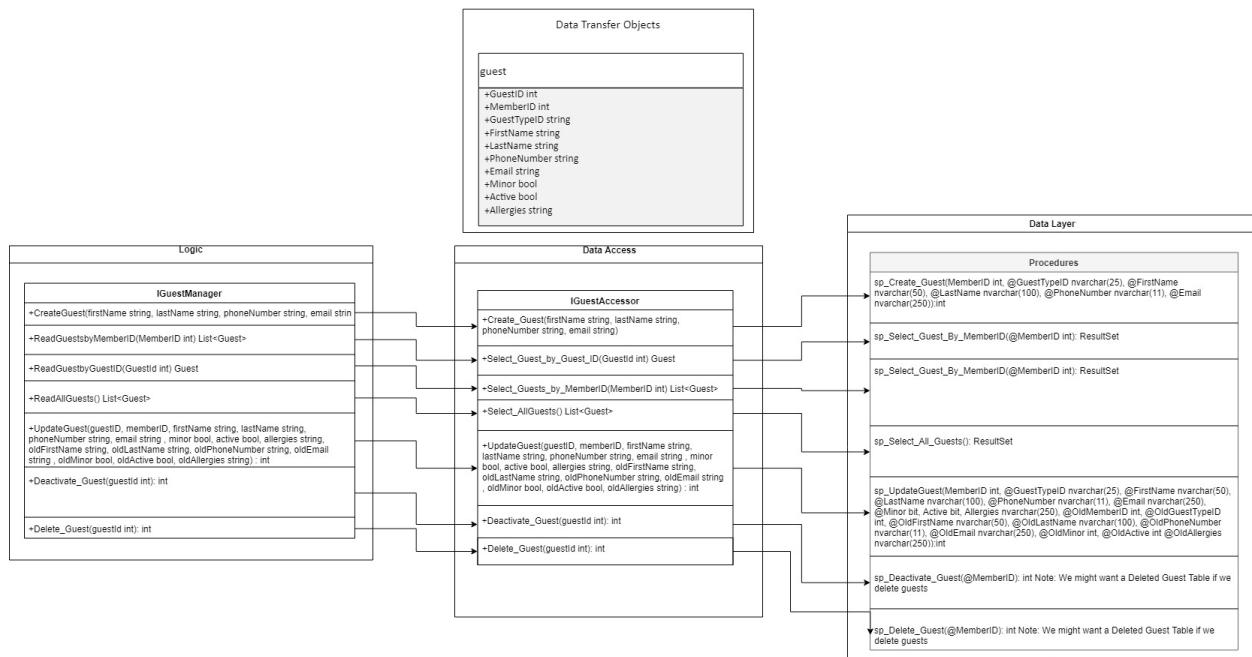
Created by: Caitlin Abelson & Carlos A.





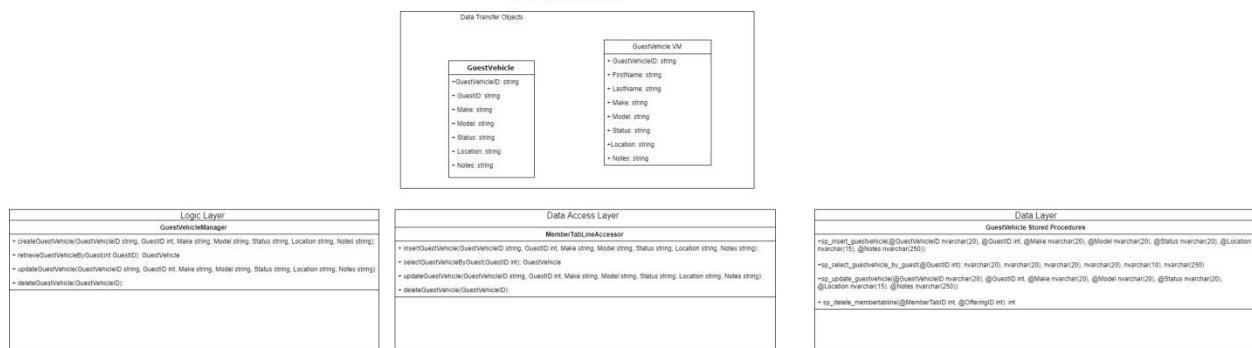
Millennial Resorts Guest Table

Created by: Eric B.

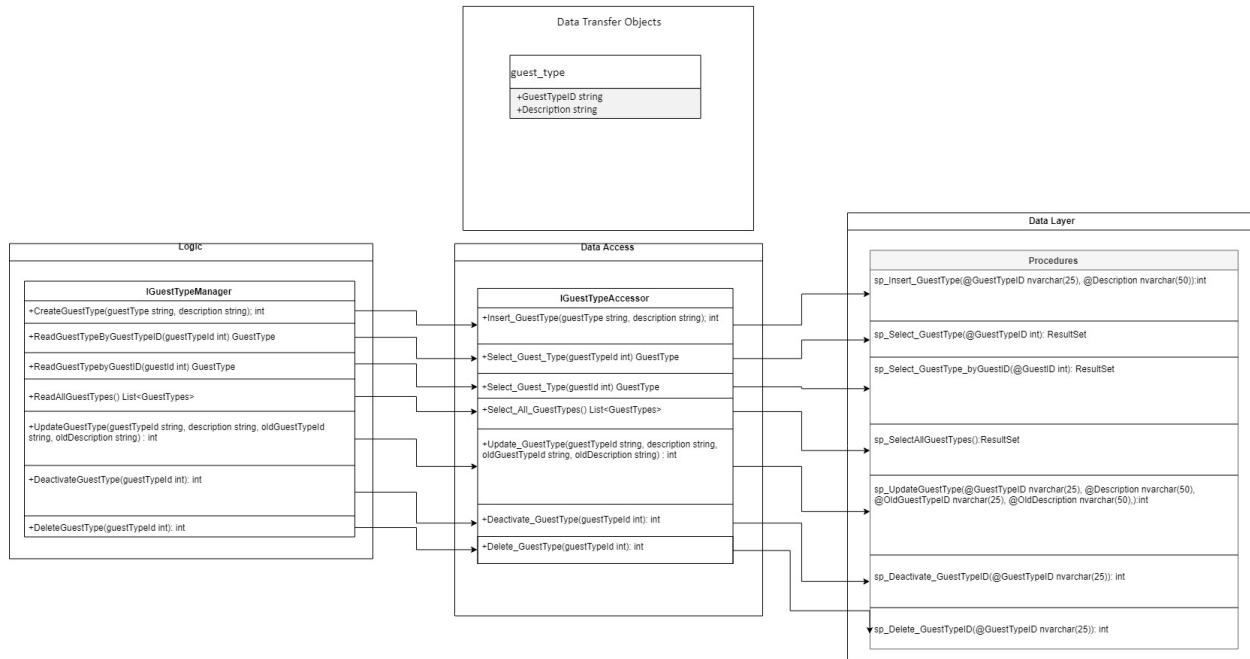


Millennial Resorts GuestVehicle Data Transfer Context Diagram

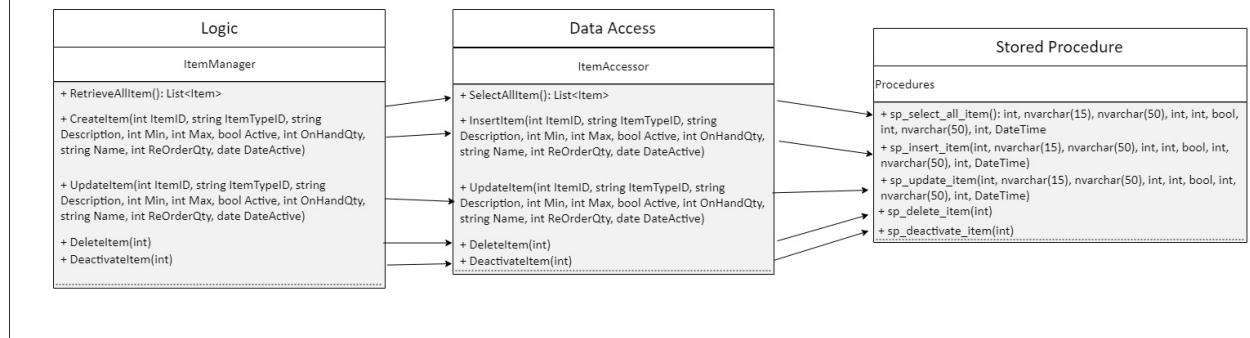
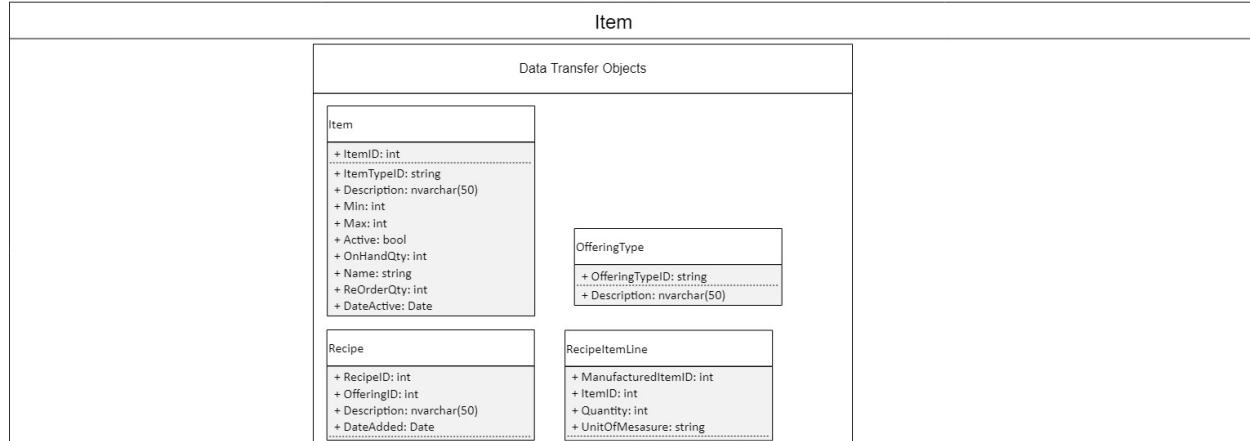
Created by: Kevin Brosko



Millennial Resorts Guest Type  
Created by: Eric B.

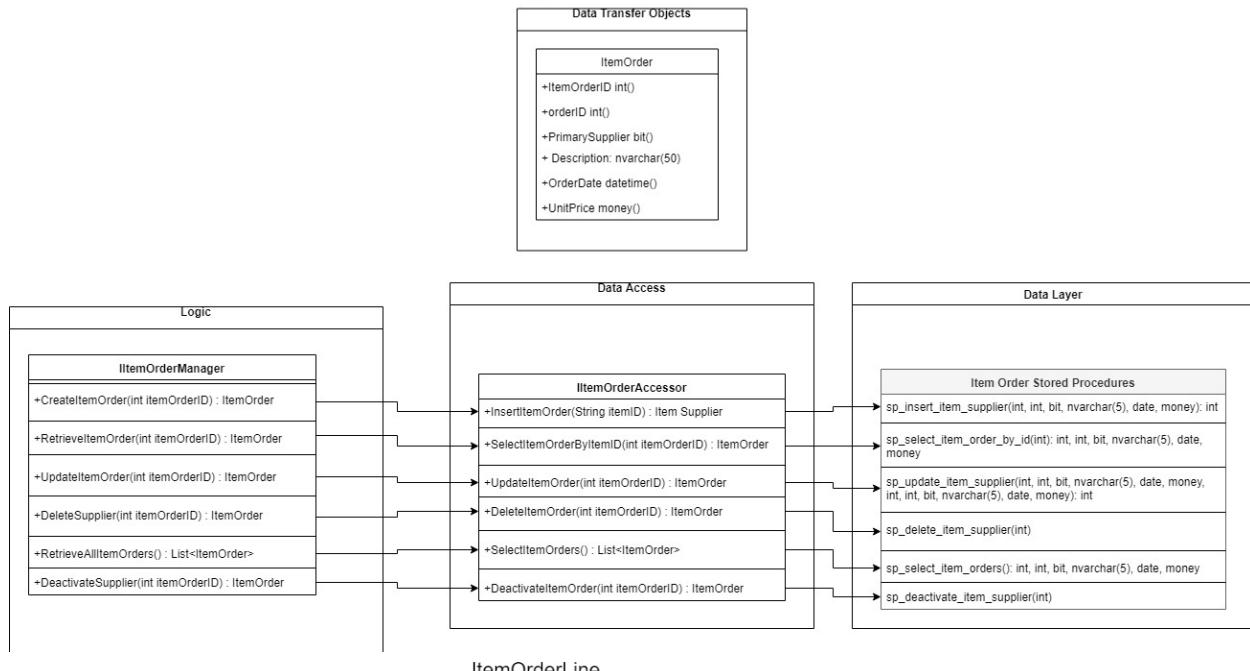


Created By:  
Ramesh Adhikari  
Edited By:  
Jared Greenfield



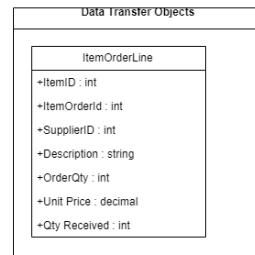
## Item Order

Created By: Chase Schultz and Caitlin Abelson



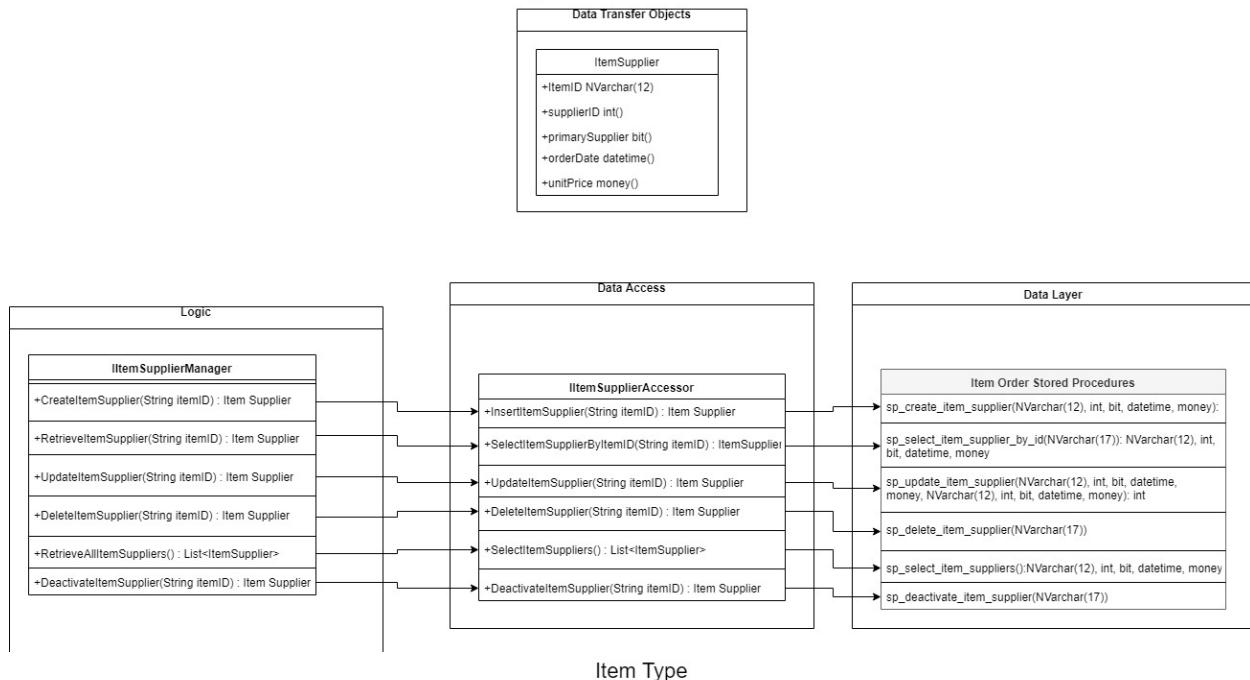
## ItemOrderLine

Created By: Richard Carroll

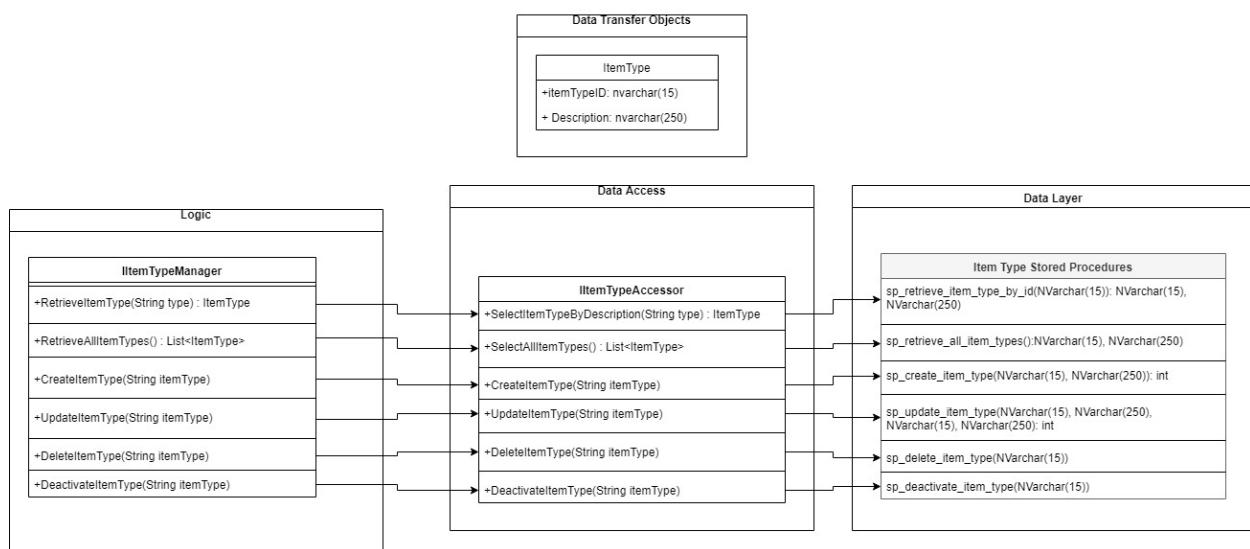


## Item Supplier

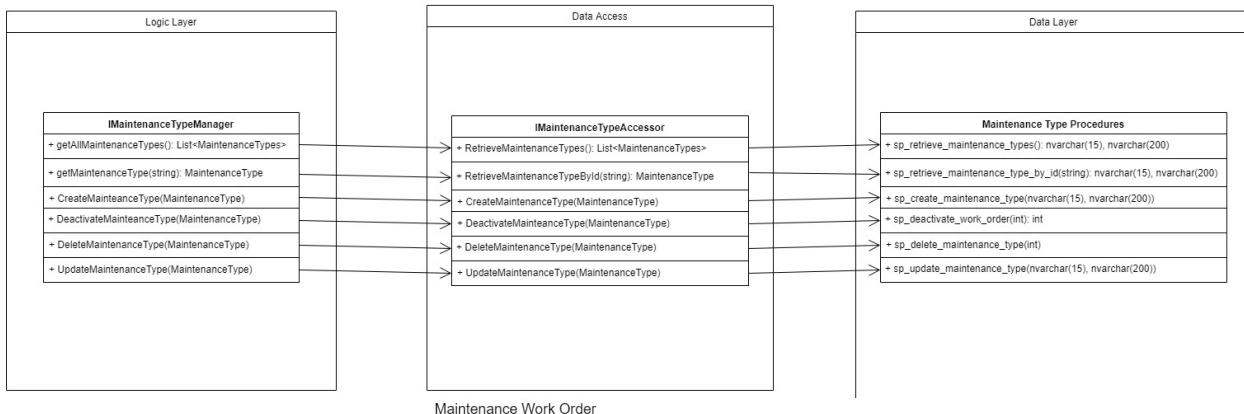
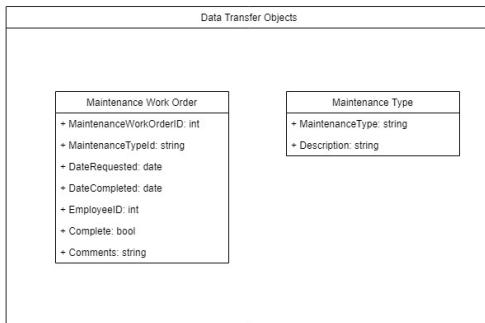
Created By: Chase Schultz and Caitlin Abelson



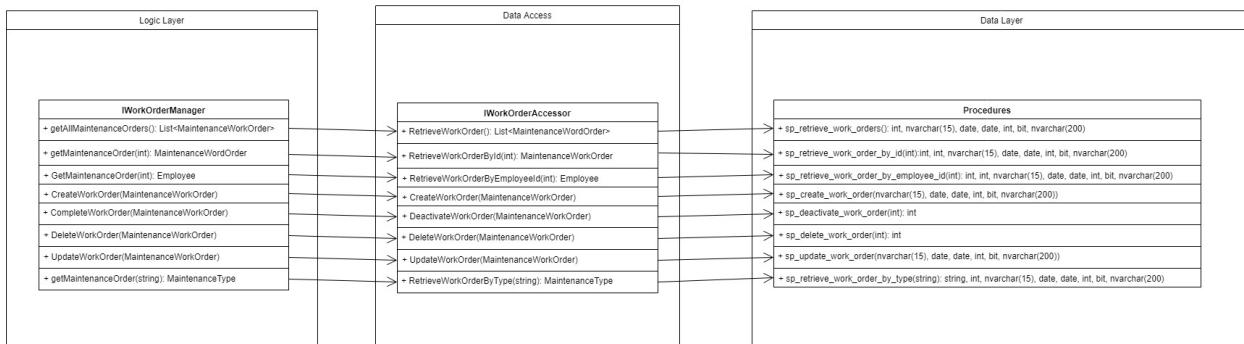
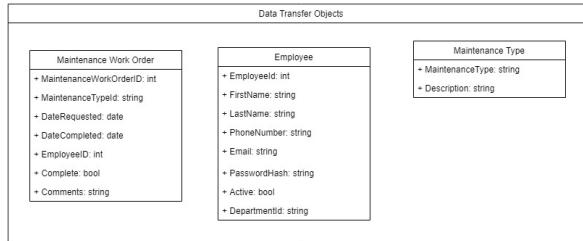
Created By: Caitlin Abelson & Jesse T



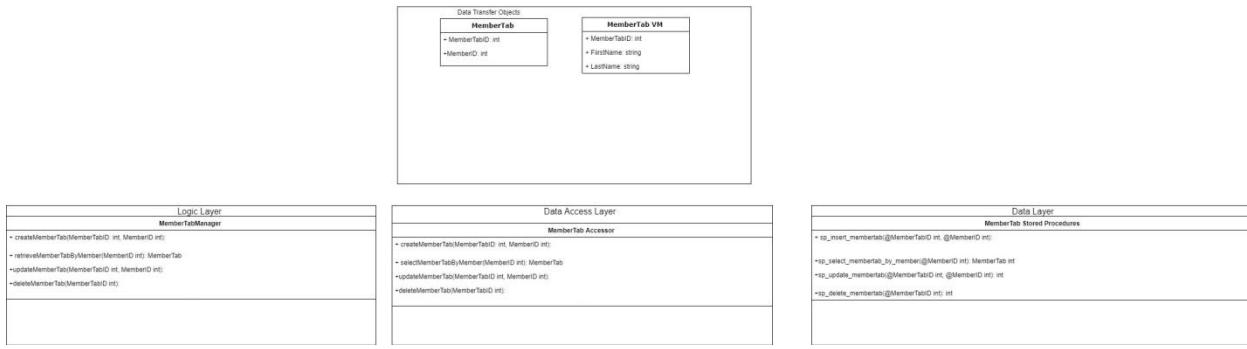
**Maintenance Type**  
Created By: Caitlin Abelson



Maintenance Work Order  
Created By: Caitlin Abelson

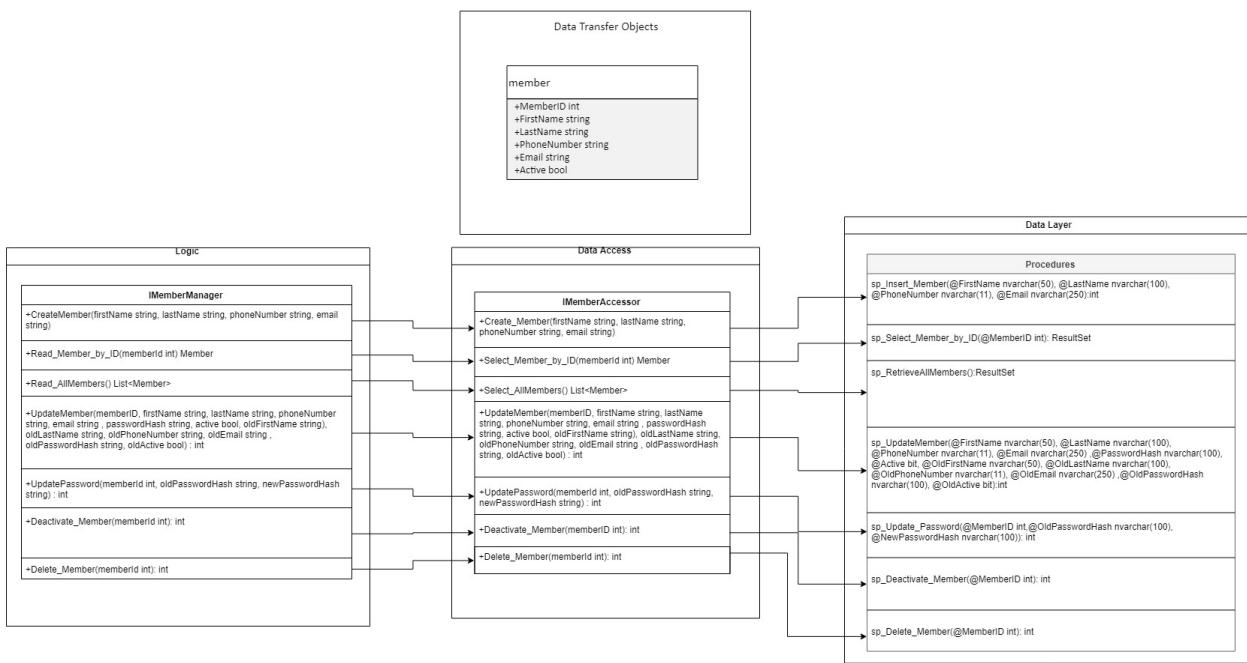


Millennial Resorts MemberTab Data Transfer Context Diagram  
Created by: Kevin Broskow



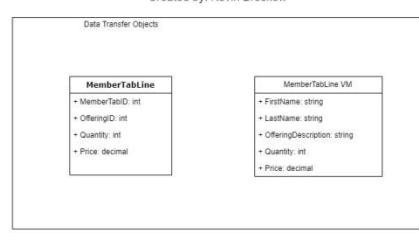
Millennial Resorts Member Table

Created by: Eric B.



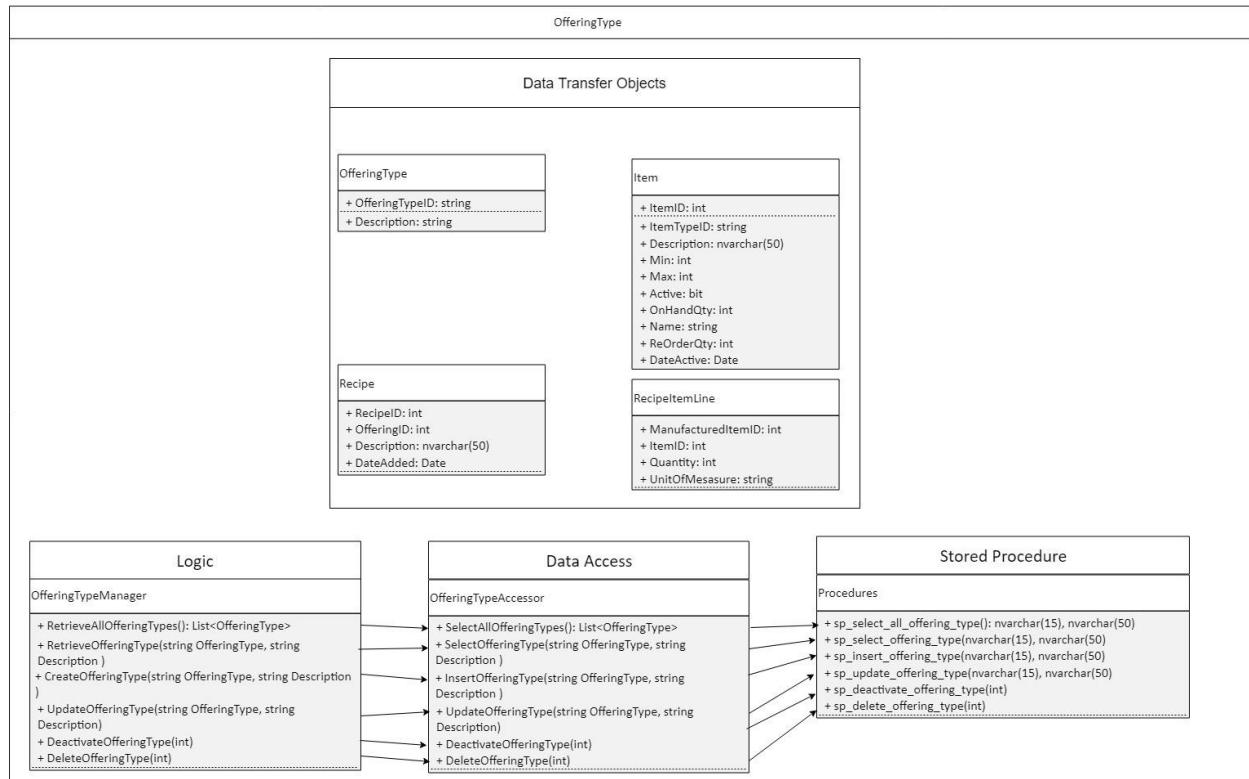
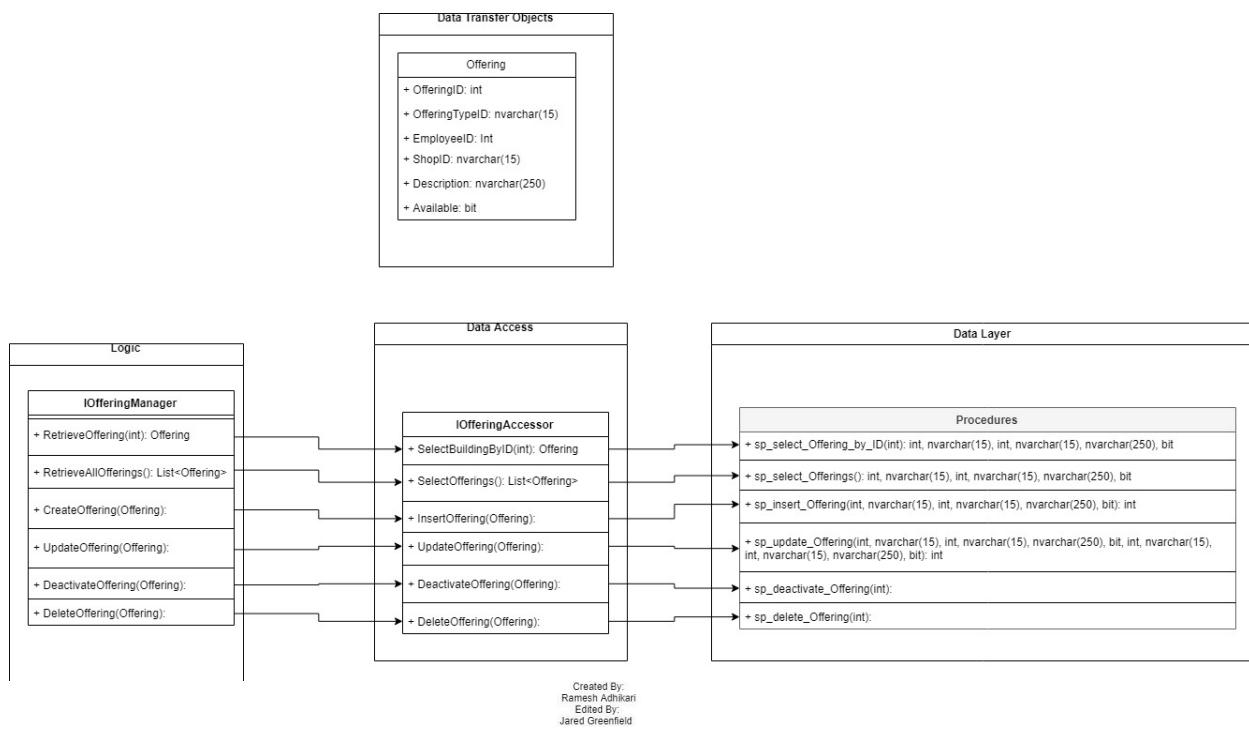
Millennial Resorts MemberTabLine Data Transfer Context Diagram

Created by: Kevin Broskow



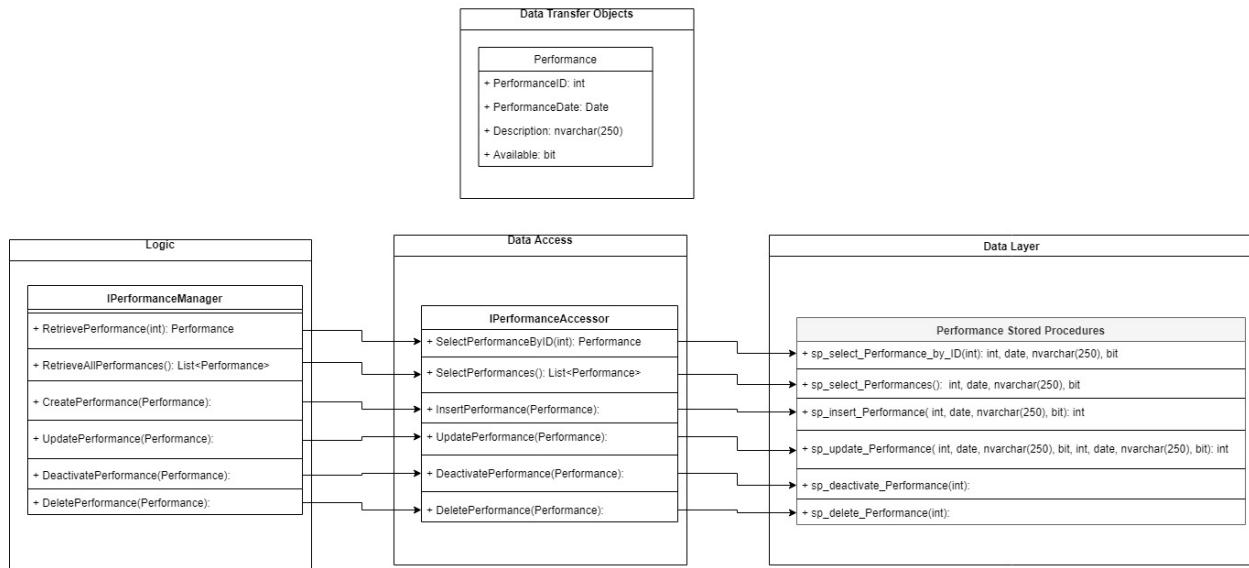
## Offering

Created By: Caitlin Abelson & Carlos A



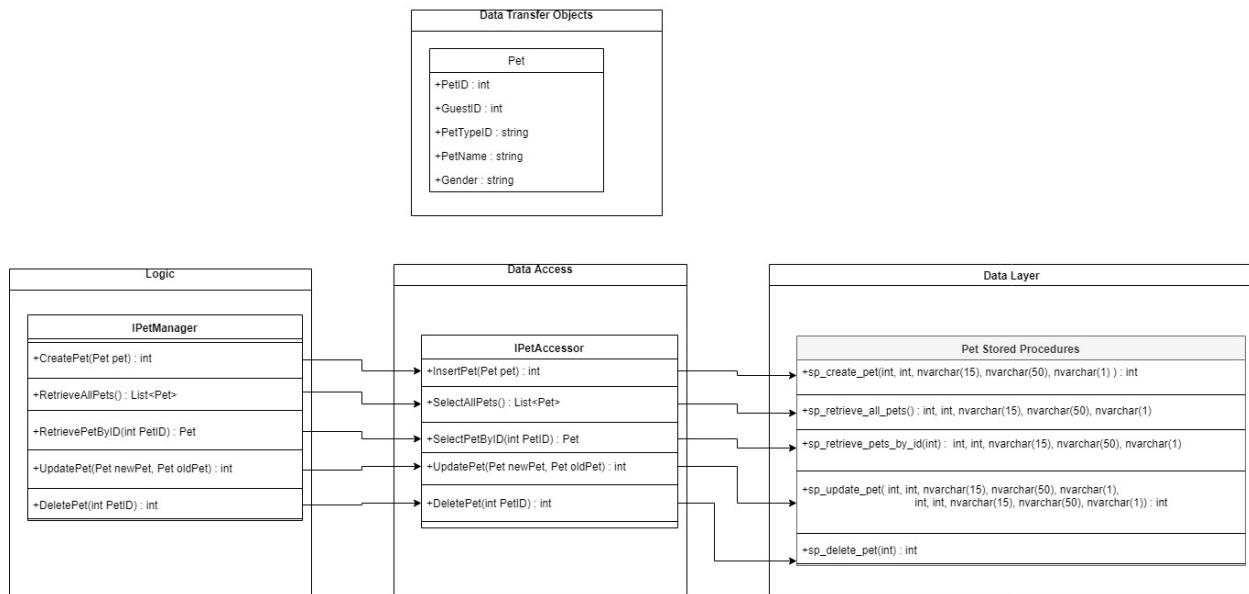
## Performance

Created By: Carlos A. and Caitlin Abelson



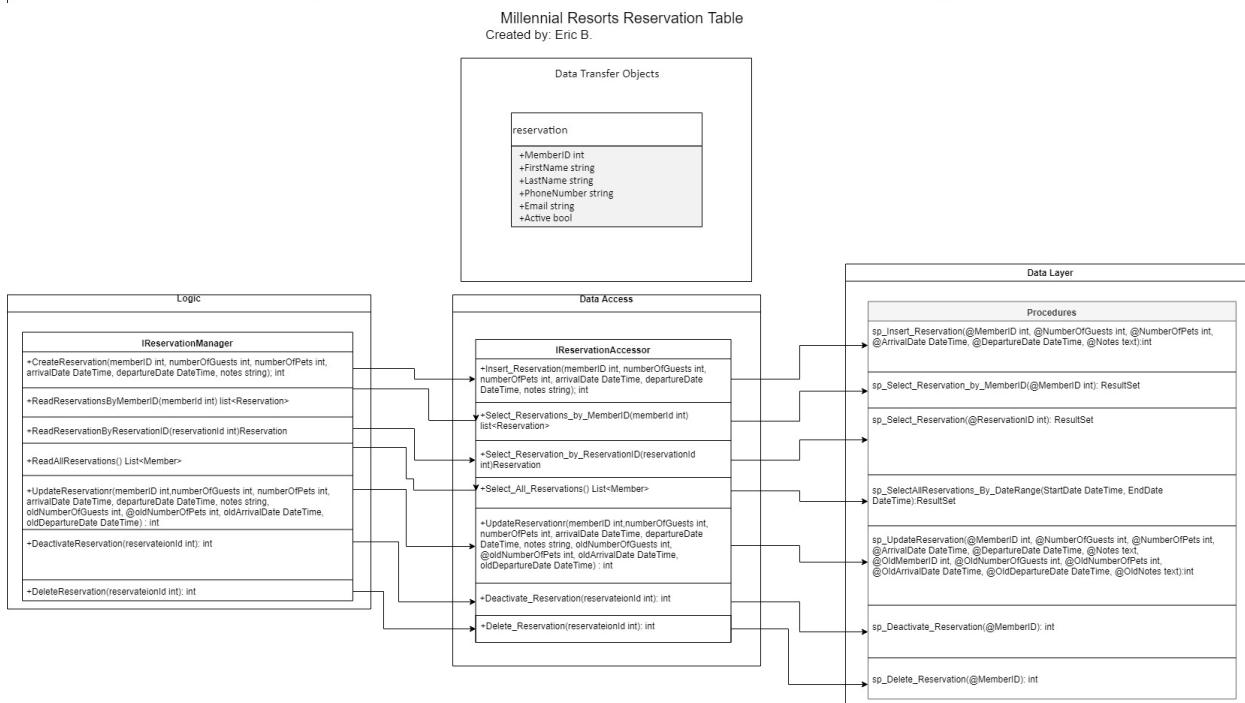
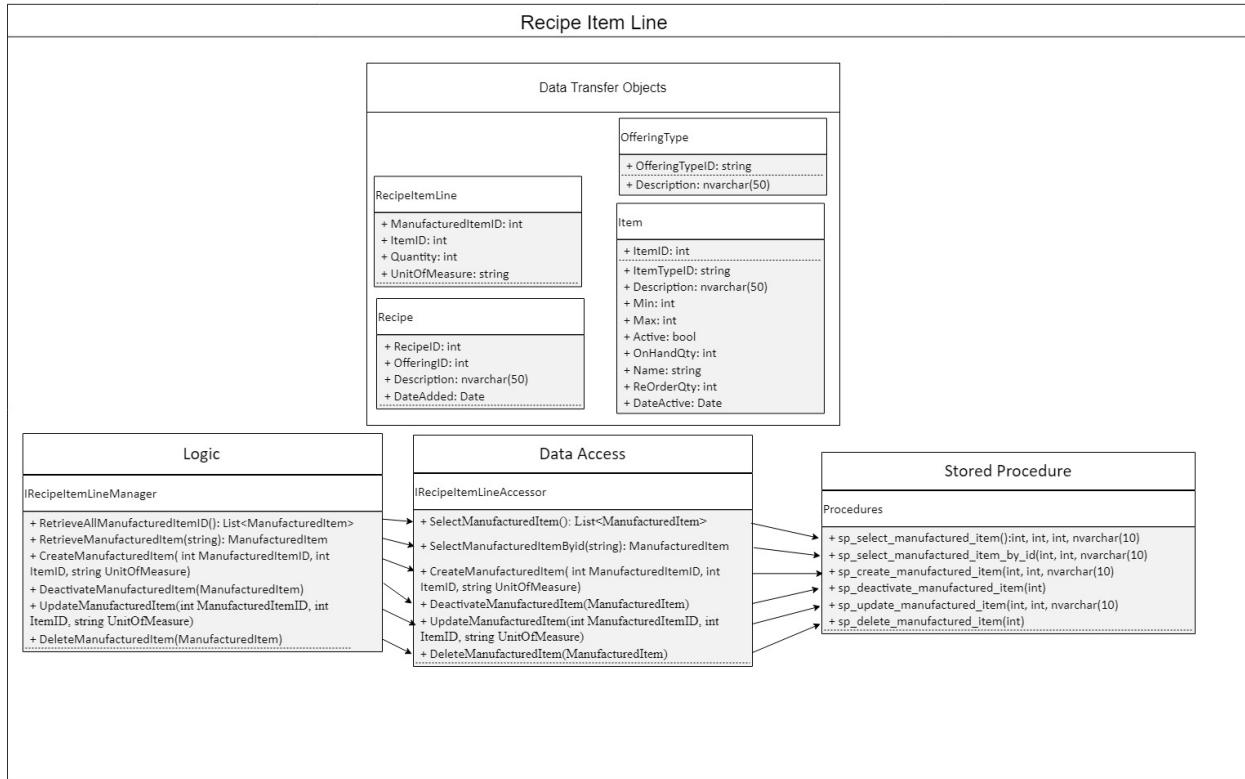
## Pet

Created By: Richard Carroll  
And Jared Greenfield

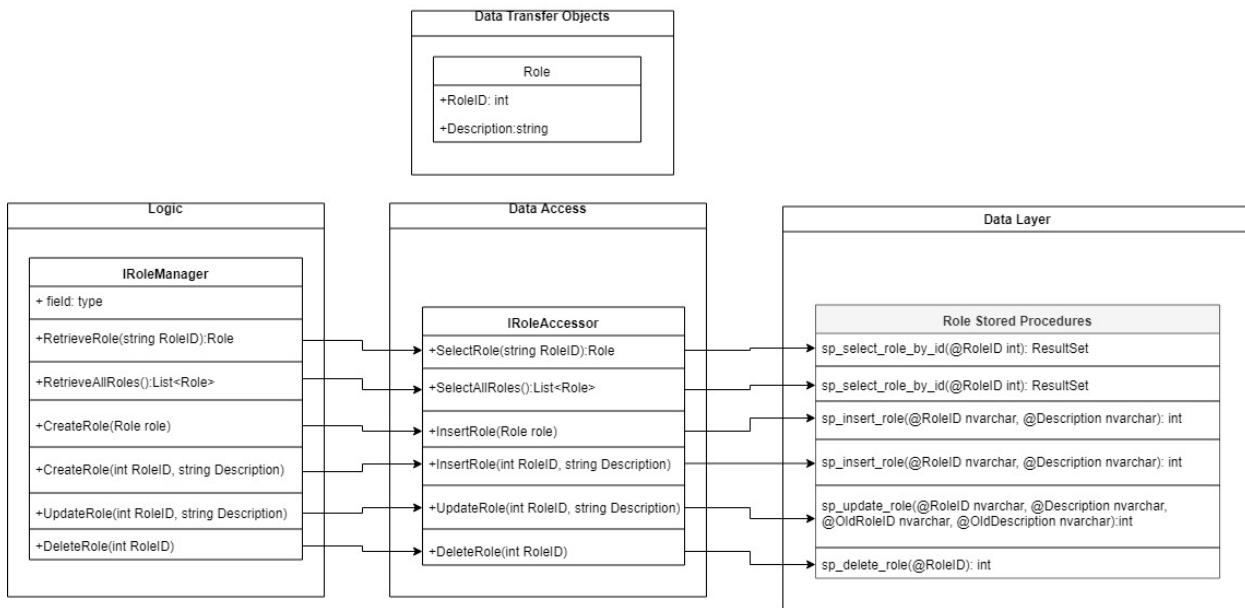




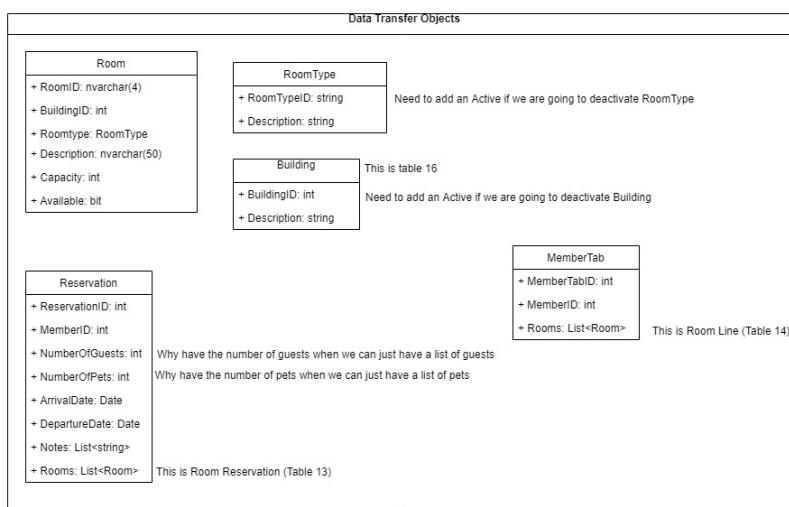
Created By:  
Ramesh Adhikari  
Edited By:  
Jared Greenfield



**Millennial Resorts Role Table**  
Created by: Jacob Miller and Caitlin Abelson

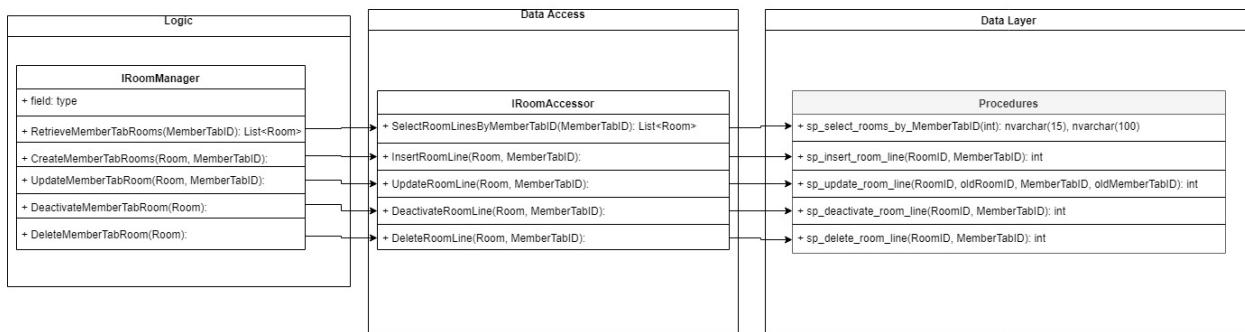


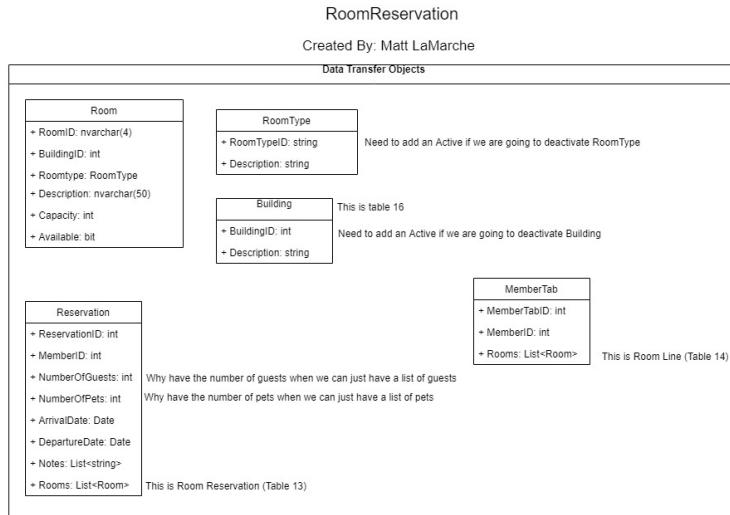
**Room Line**  
Created By: Matt LaMarche



**Room Line**

Room Line is just a List of Rooms associated with a particular Member Tab



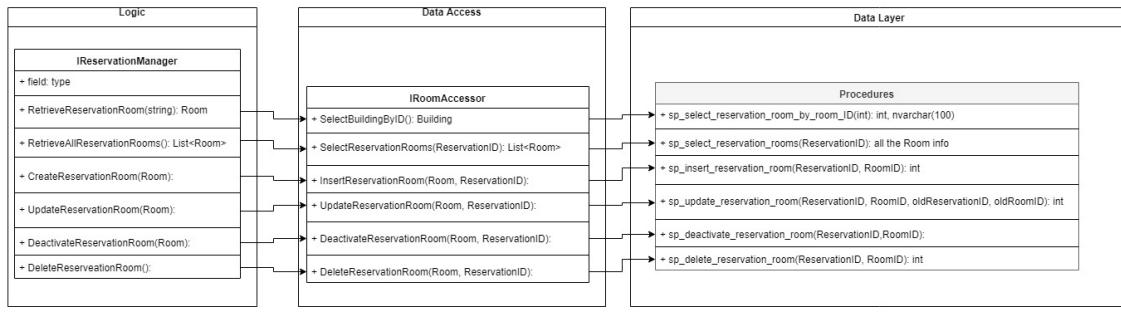


## RoomReservation

A Room Reservation is a List of Rooms associated with a Reservation.

### Tables

Room Reservation  
Room Line  
RoomType  
Building

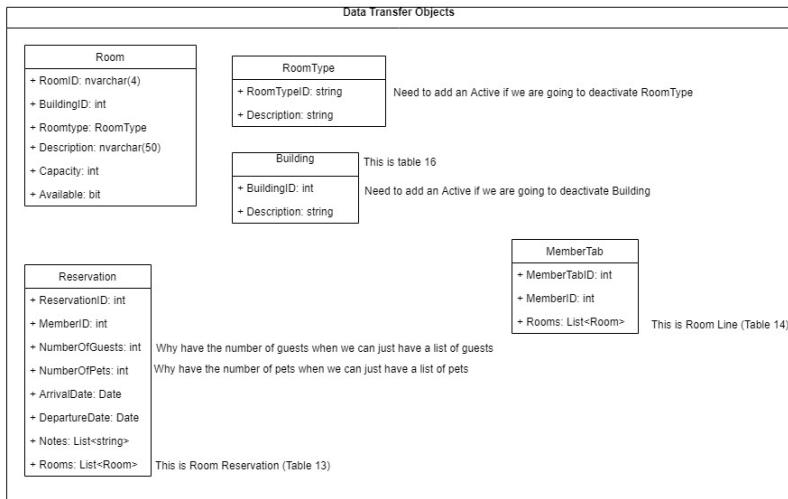


Scheduled Item  
Created By: Jared Greenfield



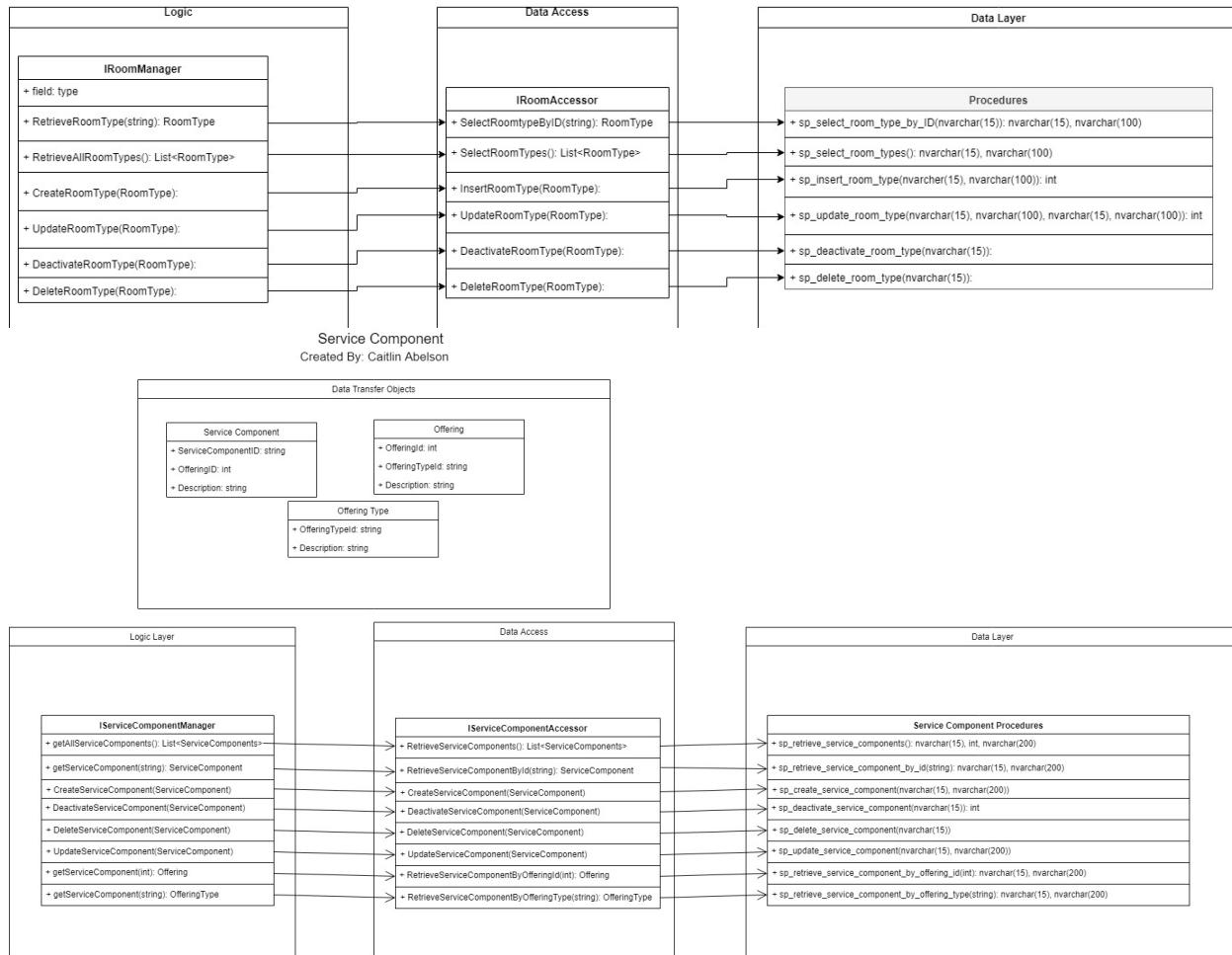
## RoomType

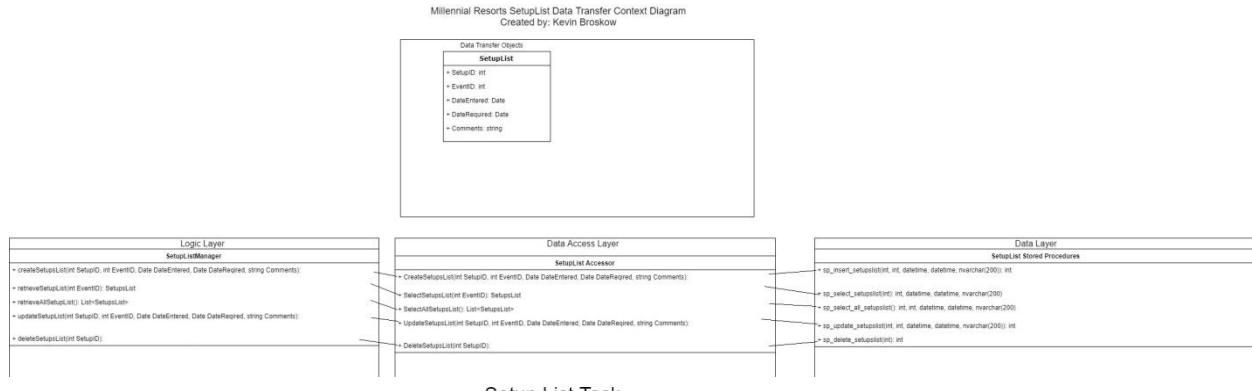
Created By: Matt LaMarche



### RoomType

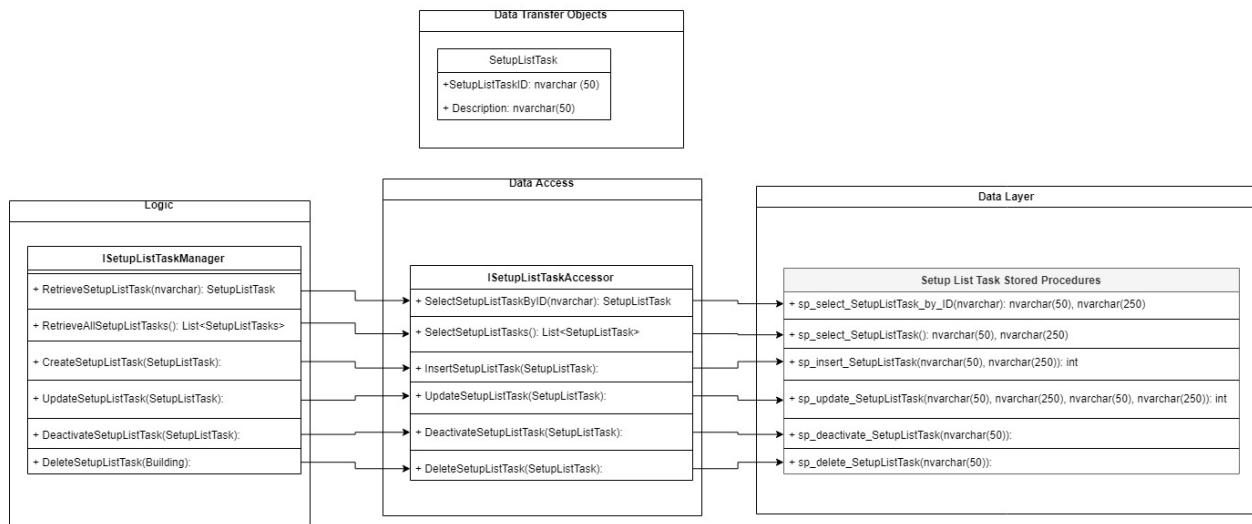
A RoomType is an Object created out of the RoomType Table. The RoomType is used in the Room Table primarily so it is implemented within the RoomManager class in the Logic Layer and the RoomAccessor Methods in the Data Access Layer.





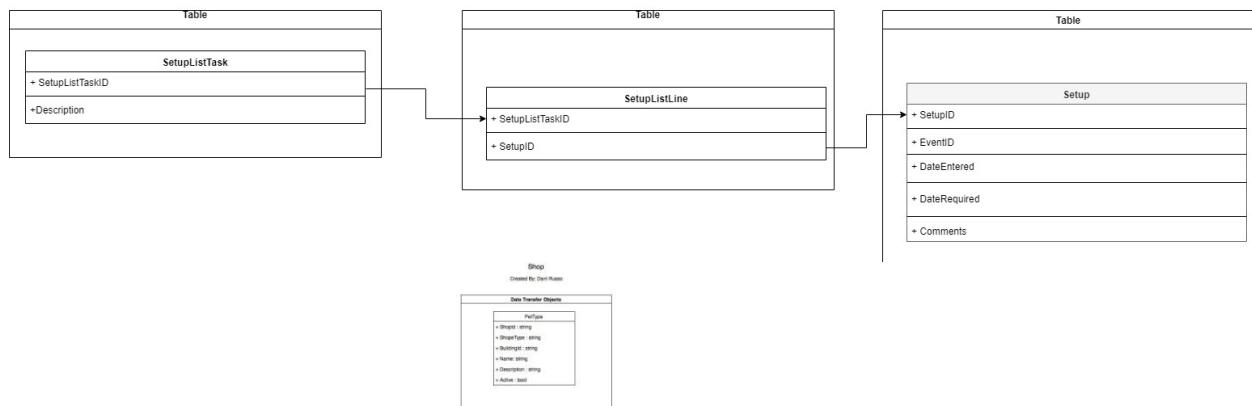
## Setup List Task

Created By: Carlos A. and Caitlin Abelson



### Setup List Line

Created by: Caitlin Abelson & Kevin E

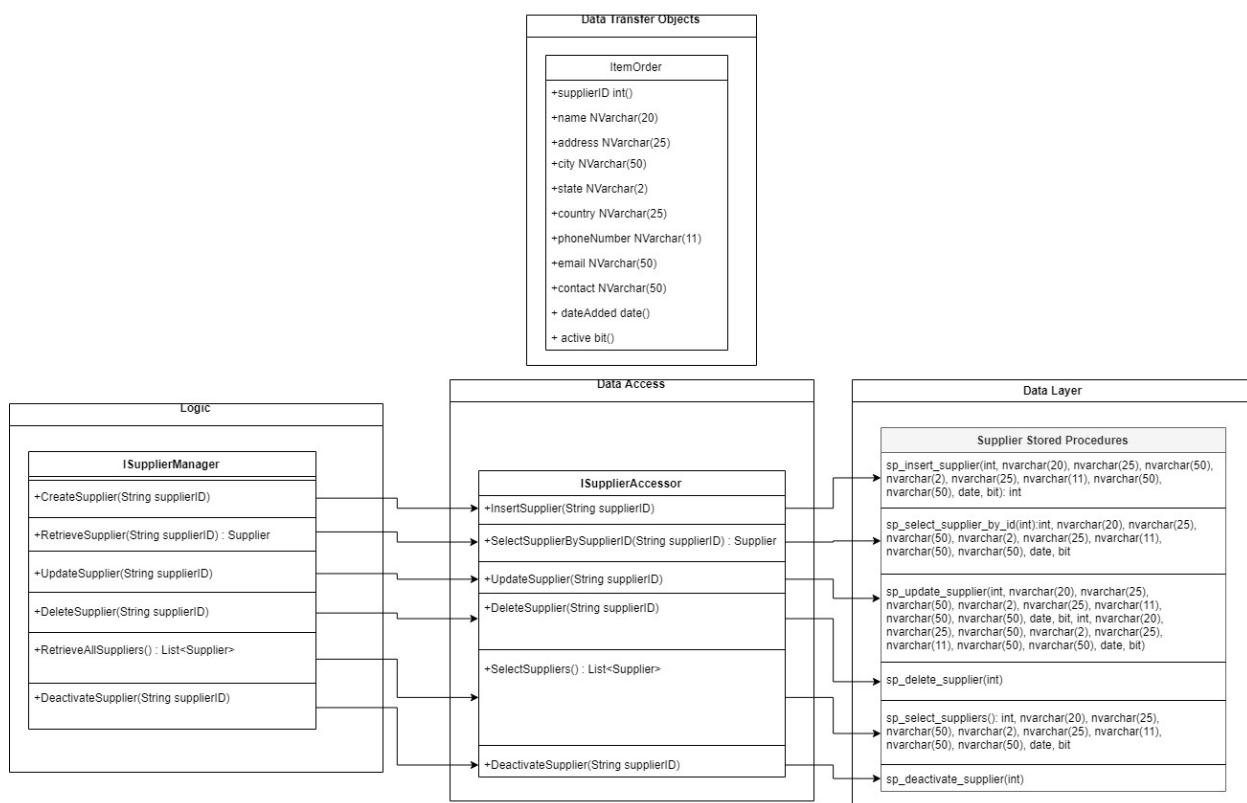


Shop



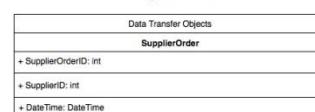
## Supplier

Created By: Chase Schultz and Caitlin Abelson



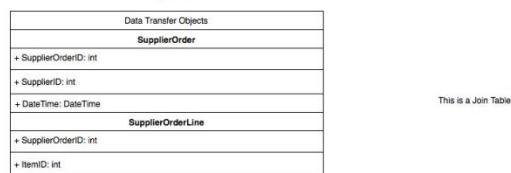
### Supplier Order

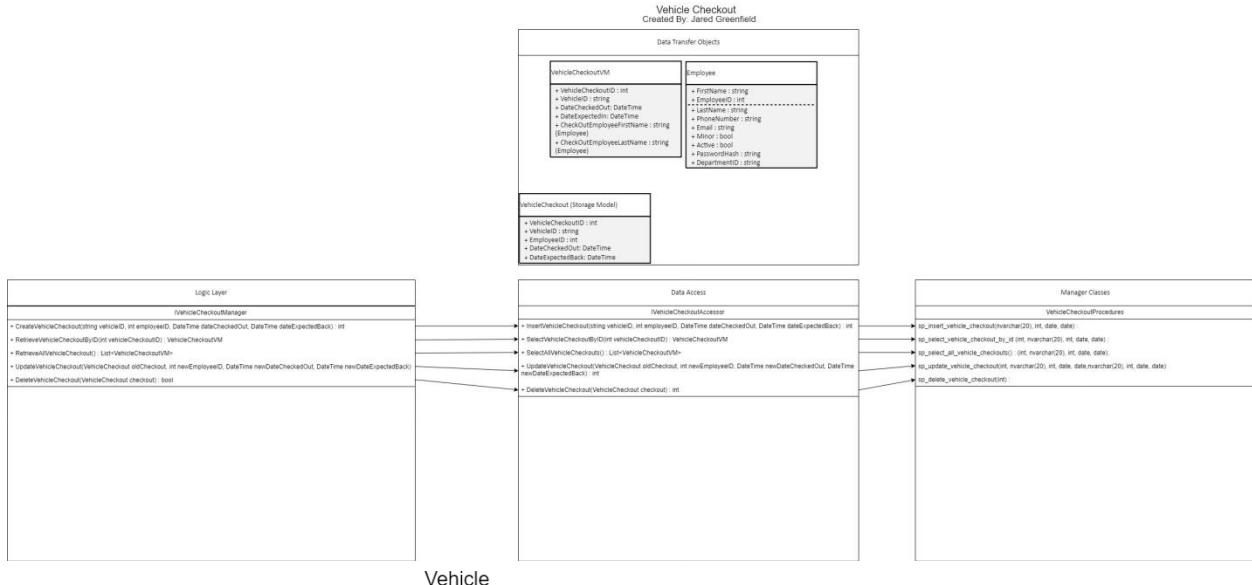
Created by: James Heim



### Supplier Order Line

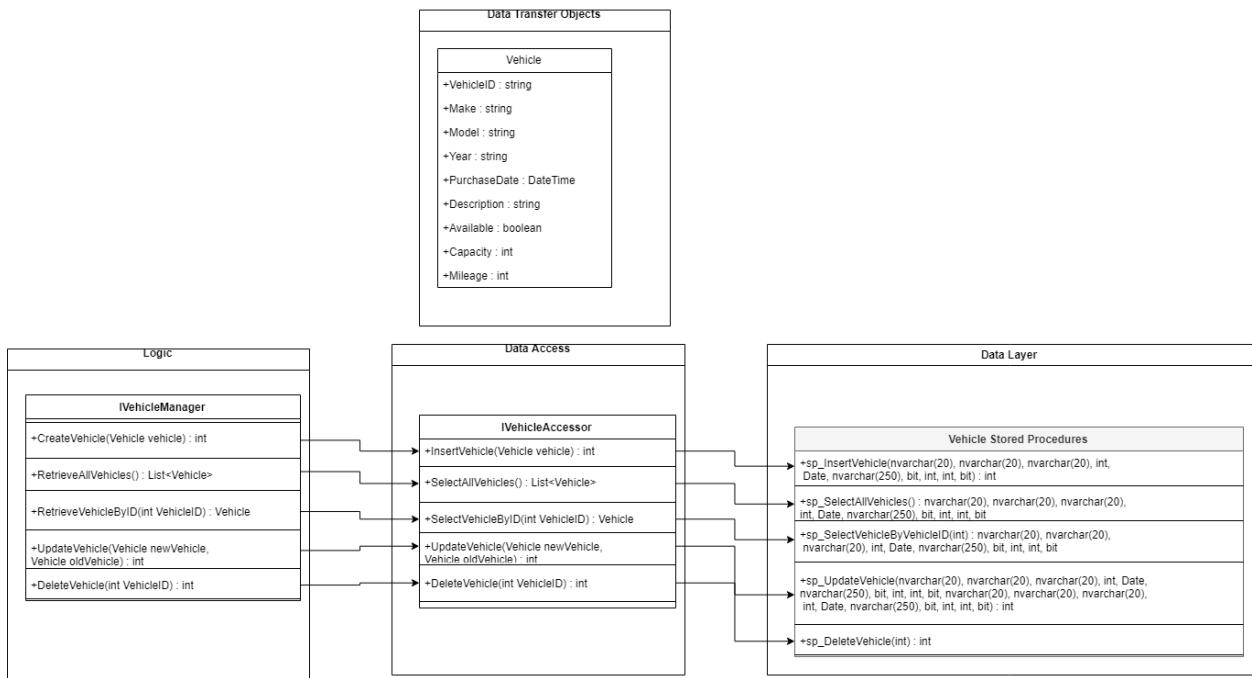
Created by: James Heim





Vehicle

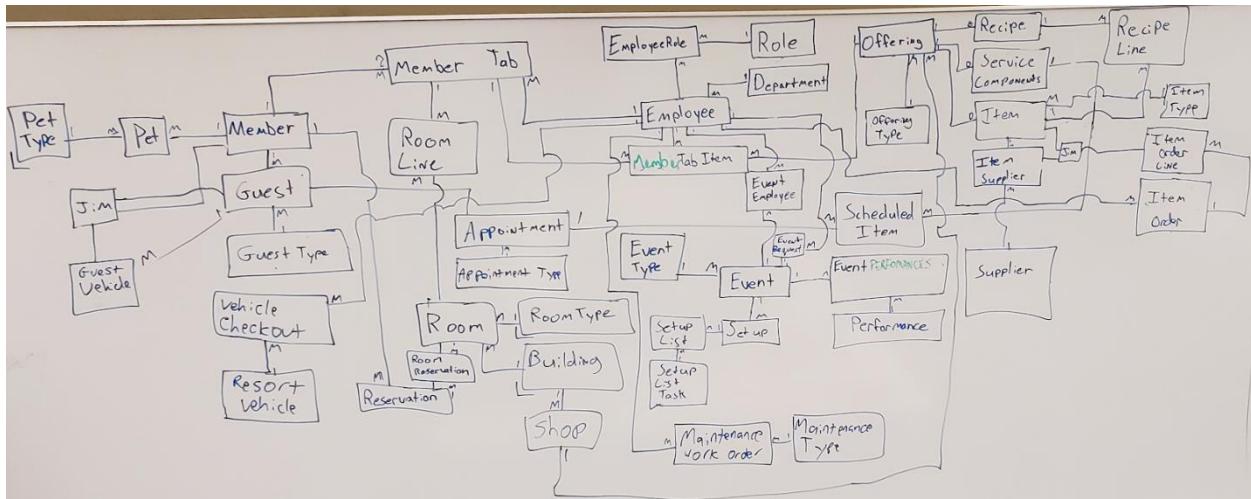
Created By: Richard Carroll



## Data Layer

### ER Diagram

The overall design of our database is best represented in the form of an ER Diagram. The ER Diagram shows both the tables we will be storing our data in as well as the relationships between these tables. Each Table will be covered in further detail in the Data Dictionary section. The ER Diagram shows a graphical depiction of our data model and the relationships between the entities. This gives a quick overview of the entire model in one view. This allows for seeing where relationships are missing or should be changed. As of now there are 48 entities defined. More will be added as the needed.



### Data Dictionary

Our Data Dictionary is a Collection of definitions of items in our Database. This Data dictionary is organized and structured in categorized tables, which has the data always available for the user. Our Data Dictionary will cover the contents, format, and structure of our database and will include elements regarding the relationships between these tables as well as the primary use of each table. The Data Dictionary shows the details of all the entities in the data layer for the purposes of our programmers and others who need to refer to them. It gives a detailed description of the items and what their purpose is. It also shows the relationships of all the entities in our data model.

### EMPLOYEE

The Employee Table holds Data regarding our employee Accounts.

Employee	
EmployeeID	int
FirstName	nvarchar

LastName	nvarchar
PhoneNumber	nvarchar
Email	nvarchar
PasswordHash	nvarchar
Active	bit
DepartmentID	nvarchar

### ROLE

The Role Table holds Data regarding the Roles our Employees can work

Role	
RoleID	nvarchar
Description	nvarchar

### EMPLOYEE ROLE

The EmployeeRole Table holds Data regarding which Roles our Employees are currently and previously able to work

EmployeeRole	
EmployeeID	int
RoleID	nvarchar

### DEPARTMENT

The Department Table holds Data regarding which Departments we have at our resort

Department	
DepartmentID	nvarchar
Description	nvarchar

### MEMBER

The Member Table holds Data regarding which Members stay at our resort

Member	
MemberID	int
FirstName	nvarchar
LastName	nvarchar

PhoneNumber	nvarchar
Email	nvarchar
PasswordHash	nvarchar
Active	bit

### GUEST

The Guest Table holds Data regarding the Guests who stay at our resort

Guest	
GuestID	int
MemberID	int
GuestTypeID	nvarchar
FirstName	nvarchar
LastName	nvarchar
PhoneNumber	nvarchar
Email	nvarchar
Minor	bit
Active	bit

### GUEST TYPE

The Guest Type Table holds Data regarding what types of Guests can stay at our resort

GuestType	
GuestTypeID	nvarchar
Description	nvarchar

### RESERVATION

The Reservation Table holds Data regarding what Reservations have been scheduled at our resort

Reservation	
ReservationID	int
MemberID	int
NumberOfGuests	int
NumberOfPets	int
ArrivalDate	Date
DepartureDate	Date
Notes	nvarchar

### MEMBER TAB

The Member Tab table holds Data regarding what offerings will get charged to which member

MemberTab	
MemberTabID	int
MemberID	int

### MEMBER TAB LINE

The Member Tab table holds Data regarding what offerings will get charged to which tab

MemberTabLine	
MemberTabID	int
OfferingID	int
Quantity	int
Price	decimal

### GUEST VEHICLE

The Guest Vehicle table holds Data regarding what Vehicles allowed to park at our resort are associated with which Member

GuestVehicle	
GuestVehicleID	nvarchar
GuestID	int
Make	nvarchar
Model	nvarchar
Status	nvarchar
Location	nvarchar
Notes	nvarchar

### ROOM

The Room table holds Data regarding what Rooms are available at our resort

Room	
RoomID	nvarchar
BuildingID	int

RoomTypeID	nvarchar
Description	nvarchar
Capacity	int
Available	bit
Active	bit

### ROOM RESERVATION

The Room Reservation table holds Data regarding what Rooms are associated with which Reservations at a specific time

RoomReservation	
RoomReservationID	int
RoomID	nvarchar
ReservationID	int
CheckInDate	DateTime
CheckOutDate	DateTime

### ROOM LINE

The Room Line table holds Data regarding what Rooms will be charged to which tab

RoomLine	
RoomID	nvarchar
MemberTabID	int
CheckInDate	DateTime
CheckOutDate	DateTime

### ROOM TYPE

The Room Type table holds Data regarding what Type of Rooms our resort has to offer

RoomType	
RoomTypeID	nvarchar
Description	nvarchar
Active	bit

### BUILDING

The Building table holds Data regarding which Buildings our resort has access to

## Building

BuildingID	nvarchar
Description	nvarchar
Active	bit

## MAINTENANCE WORK ORDER

The Maintenance Work Order table holds Data regarding what Maintenance work needs to be done where as well as which Employee requested the Maintenance work and Which Employee was assigned to do the work.

## MaintenanceWorkOrder

MaintenanceWork	int
OrderID	
MaintenanceType	nvarchar
D	
DateRequested	Date
DateCompleted	Date
EmployeeID	int
Complete	bit
Comments	nvarchar

## MAINTENANCE TYPE

The Maintenance Type table holds Data regarding what Types of Maintenance Work our Resort can perform.

## MaintenanceType

MaintenanceType	nvarchar
D	
Description	nvarchar
Active	bit

## SERVICE COMPONENTS

The Service Components table holds Data regarding which Services our resort offers to Guests and Members

## ServiceComponents

ServiceComponentID	nvarchar
OfferingID	int
Description	nvarchar
Active	bit

### EVENT

The Event table holds Data regarding which Events are scheduled as well as which Employees are assigned to work at the Event

Event	
EventID	int
EventTypeID	nvarchar
EventRequestID	int
EventDate	DateTime
Description	nvarchar
KidsAllowed	bit

### EVENT TYPE

The Event Type table holds Data regarding what types of Events our Resort offers

EventType	
EventTypeID	nvarchar
Description	nvarchar
Active	bit

### EVENT REQUEST

The Event Request table holds Data regarding the details of who, what, when, and where the event will take place as well as who requested the Event.

EventRequest	
EventRequestID	int
EmployeeID	int
EventTypeID	nvarchar
Description	nvarchar
EventRequestDate	DateTime
EventDate	DateTime

KidsAllowed	bit
NumGuests	int
Location	nvarchar

### SETUP LISTS

The Setup List table holds Data regarding which tasks are required to set up an Event

SetupLists	
SetupID	int
EventID	int
DateEntered	Datetime
DateRequired	Datetime
Comments	nvarchar

### SETUP LIST LINE

The Setup List Line table holds Data regarding which tasks are associated with which Setup List

SetupListLine	
SetupID	int
SetupListTaskID	nvarchar

### SETUP LIST TASKS

The Setup List Tasks table holds Data regarding which tasks are available to go on a Setup List

SetupListTasks	
SetupListTaskID	nvarchar
Description	nvarchar

### PERFORMANCE

The Performance table holds Data regarding which Performances the resort offers

Performance	
PerformanceID	int
PerformanceDate	Datetime
Description	nvarchar
Active	bit

### EVENT PERFORMANCE

The Event Performance table holds Data regarding which Performances are going to be played at which Event

EventPerformance	
PerformanceID	int
EventID	int

### OFFERING

The Offering table holds Data regarding which Services, Items, and charges have been charged to a Member

Offering	
OfferingID	int
OfferingTypeID	nvarchar
EmployeeID	int
ShopID	nvarchar
Description	nvarchar
Active	bit

### OFFERING TYPE

The Offering Type table holds Data regarding what types of Offerings our resort offers

OfferingType	
OfferingTypeID	nvarchar
Description	nvarchar

### RECIPE

The Recipe table holds Data regarding which Items resort sells which the resort also makes

Recipe	
RecipeID	int
OfferingID	int
Description	nvarchar
DateAdded	DateTime

Active	bit
--------	-----

#### RECIPE ITEM LINE

The Recipe Item Line holds Data regarding which Items go into which Recipe

RecipeItemLine	
RecipeID	int
ItemID	int
Quantity	int
UnitofMeasure	nvarchar

#### ITEM

The Item table holds Data regarding which Items are consumed and sold by our resort

Item	
ItemID	int
ItemTypeID	nvarchar
Description	nvarchar
Min	int
Max	int
OnHandQty	int
Name	nvarchar
ReOrderQty	int
DateActive	DateTime
Price	Money
Active	bit

#### ITEM TYPE

The Item type table holds Data regarding which types of Items are available at our resort

ItemType	
ItemTypeID	nvarchar
Description	nvarchar

#### SUPPLIER

The Supplier table holds Data regarding which Suppliers our resort orders supplies from

Supplier	
SupplierID	int
Name	nvarchar
Address	nvarchar
City	nvarchar
State	nvarchar
Country	nvarchar
PhoneNumber	nvarchar
Email	nvarchar
Contact	nvarchar
DateAdded	DateTime
Active	bit

#### ITEM SUPPLIER

The Item Supplier Table holds Data regarding which Suppliers provide which Item at our resort

ItemSupplier	
ItemID	nvarchar
SupplierID	int
PrimarySupplier	bit
OrderDate	datetime
UnitPrice	money

#### ITEM ORDER

The Item Order table holds Data regarding Orders for Items have been requested by an Employee

ItemOrder	
ItemOrderID	int
EmployeeID	int
Description	nvarchar
OrderComplete	bit
DateOrdered	DateTime

#### ITEM ORDER LINE

The Item Order Line table holds Data regarding which Item was requested by which Order

### ItemOrderLine

ItemID	int
ItemOrderID	int
SupplierID	int
Description	nvarchar
OrderQty	int
Unit Price	money
QtyReceived	int

### PET

The Pet table holds Data regarding which Pets are associated with which Members at our resort

### Pet

PetID	int
GuestID	int
PetTypeID	nvarchar
PetName	nvarchar
Gender	nvarchar

### RESORT VEHICLE

The Resort Vehicle table holds Data regarding which vehicles are available for Employee use at our resort

### ResortVehicle

VehicleID	nvarchar
Make	nvarchar
Model	nvarchar
Year	int
PurchaseDate	Date
Description	nvarchar
Available	bit
Capacity	int
Mileage	int
Active	bit

### VEHICLE CHECKOUT

The Vehicle Checkout table holds Data regarding which Vehicles were checked out to which Guest and when

VehicleCheckout	
VehicleCheckoutID	int
ResortVehicleID	nvarchar
EmployeeID	int
DateCheckedOut	DateTime
DateReturned	DateTime
DateExpectedBack	DateTime

### SCHEDULED ITEM

The Scheduled Item table holds Data regarding which Items used at an Appointment

ScheduledItem	
AppointmentID	int
ServiceComponentID	nvarchar

### APPOINTMENT

The Appointment table holds Data regarding which Appointments are attended by and charged to which Member or Guest

Appointment	
AppointmentID	int
AppointmentTypeID	nvarchar
GuestID	int
DateTime	DateTime
Description	nvarchar

### APPOINTMENT TYPE

The Appointment Type table holds Data regarding which types of Appointments are available at our resort

AppointmentType	
-----------------	--

AppointmentTypeID	
D	nvarchar
Description	nvarchar
Active	bit

### SHOP

The Shop table holds Data regarding which Shops are available at our resort

Shop	
ShopID	nvarchar
BuildingID	nvarchar
Name	nvarchar
Description	nvarchar
Active	bit

### PET TYPE

The Pet Type table holds Data regarding which types of Pets our resort will cater to

PetType	
PetTypeID	nvarchar
Description	nvarchar
Species	nvarchar

### EVENT EMPLOYEE

The Event Employee Table holds Data regarding which Events are scheduled to work at which Event

EventEmployee	
EmployeeID	int
EventID	int

### SUPPLY ORDER

The Supply Order table holds Data regarding which Supply Orders have been sent to which Supplier

SupplyOrder	
SupplierOrderID	int
SupplierID	int

DateTime	DateTime
----------	----------

### SUPPLY ORDER LINE

The Supply Order Line table holds Data regarding which Items were associated with which Supply Order

SupplyOrderLine	
SupplierOrderID	int
ItemID	int

## Stored Procedures

Stored procedures are a set of instructions to perform a specific task and obtain a result or results from the database required by another task or user. They are stored and are always at the disposal of the user. In order to easily and efficiently access the data stored in our database we have developed these stored procedures to further simplify data access and data modification. Following is a list of all the stored procedures we will be using to develop the solution for the resort. No plain text queries of the data layer will be allowed. The use of stored procedures gives us a secure interface to the database and also makes it easier to make changes to data layer access without having to recompile the program.

```
sp_select_room_by_id(id)
sp_select_rooms()
sp_insert_room(Details)
sp_update_room(new room, old room)
sp_deactivate_room(id)
sp_delete_room(id)
```

```
sp_select_member_tab_line_by_member_tab_line_id(id)
sp_select_member_tab_line()
sp_insert_member_tab_line(details)
sp_update_member_tab_line(old member tab, new member tab)
sp_deactivate_member_tab(id)
sp_delete_member_tab(id)
```

```
sp_select_guest_vehicle_by_vehicle_id(id)
sp_select_guest_vehicles()
sp_insert_guest_vehicle(vehicle)
sp_update_guest_vehicle(old vehicle, new vehicle)
sp_deactivate_guest_vehicle(id)
sp_delete_guest_vehicle(id)
```

```
sp_insert_vehicle_checkout(vehicle checkout)
sp_select_vehicle_checkout_by_id(id)
sp_select_all_vehicle_checkout()
sp_update_vehicle_checkout(old vehicle checkout, new vehicle checkout)
sp_delete_vehicle_checkout(id)
```

```
sp_insert_reservation(reservation)
sp_select_reservation_by_memeber_id(id)
sp_select_reservation_by_reservation_id(id)
sp_select_all_reservations_by_date_range(date range)
sp_update_reservation(old reservation, new reservation)
sp_delete_reservation(id)
```

```
sp_insert_member(member)
sp_select_member_by_id(id)
sp_update_member(old member, new member)
sp_update_password(old password, new password)
sp_delete_member(id)
```

```
sp_insert_guest_type(guest type)
sp_select_guest_type_by_guest_type_id(id)
```

```
sp_select_guest_type_by_guest_id(id)
sp_select_all_guest_types()
sp_update_guest_type(old guest type, new guest type)
sp_delete_guest_type(id)
```

```
sp_create_guest(guest)
sp_select_guest_by_member_id(id)
sp_select_guest_by_guest_id(id)
sp_select_all_guests()
sp_update_guest(old guest, new guest)
sp_delete_guest(id)
```

```
sp_select_room_type_by_id(id)
sp_select_all_room_types()
sp_insert_room_type(room type)
sp_update_room_type(old room type, new room type)
sp_deactivate_room_type(id)
sp_delete_room_type(id)
```

```
sp_select_reservation_room_by_id(id)
sp_select_all_reservation_rooms()
sp_insert_reservation_room(reservation room)
sp_deactivate_reservation_room(id)
sp_delete_reservation_room(id)
sp_update_reservation_room(old reservation room, new reservation room)
```

```
sp_select_room_lines_by_member_tab_id(id)
sp_insert_room_line(room line)
sp_delete_room_line(id)
```

```
sp_deactivate_room_line(id)  
sp_update_room_line(old room line, new room line)
```

```
sp_insert_role(role)  
sp_select_role_by_id(id)  
sp_update_role(old role, new role)  
sp_delete_role(id)
```

```
sp_insert_employee(employee)  
sp_update_employee(old employee, new employee)  
sp_delete_employee(id)
```

```
sp_insert_department(department)  
sp_select_department_by_id(id)  
sp_select_department_by_description(search term)  
sp_update_department(old department, new department)  
sp_delete_department(id)
```

```
sp_select_all_work_orders()  
sp_select_work_order(id)  
sp_select_work_order_by_employee_id(id)  
sp_select_work_order_by_type(type)  
sp_insert_work_order(work order)  
sp_update_work_order(old work, new order)  
sp_deactivate_work_order(work order)  
sp_delete_work_order(id)
```

```
sp_select_all_maintenance_types()  
sp_select_maintenance_type_by_id()
```

```
sp_insert_maintenance_type(maintenance type)
sp_update_maintenance_type(old maintenance type, new maintenance type)
sp_deactivate_maintenance_type(id)
sp_delete_maintenance_type(id)
```

```
sp_select_all_buildings()
sp_select_building_by_id(id)
sp_insert_building(building)
sp_update_building(old building, new building)
sp_deactivate_building(id)
sp_delete_building(id)
```

```
sp_insert_setup_list_task(setup list task)
sp_select_setup_list_task_by_id(id)
sp_select_all_setup_list_tasks()
sp_update_setup_list_task(old setup list task, new setup list task)
sp_deactivate_setup_list_task(id)
sp_delete_setup_list_task(id)
```

```
sp_insert_setup_list(setup list)
sp_select_all_setup_list()
sp_select_setup_list_by_id(id)
sp_update_setup_list(old setup list, new setup list)
sp_deactivate_setup_list(id)
sp_delete_setup_list(id)
sp_authenticate_event(id)
```

```
sp_insert_service_component(service_component)
sp_select_all_service_components()
```

```
sp_select_service_component_by_id(id)
sp_select_service_component_by_offering_id(id)
sp_select_service_component_by_offering_type(type)
sp_update_service_component(old service component, new service component)
sp_deactivate_service_component(id)
sp_delete_service_component(id)
```

```
sp_insert_event_type(event_type)
sp_select_all_event_type()
sp_select_event_type_by_id(id)
sp_update_event_type(old event type, new event type)
sp_deactivate_event_type(id)
sp_delete_event_type(id)
```

```
sp_insert_event_request(event request)
sp_select_all_event_request()
sp_select_event_request_by_date(datetime)
sp_select_event_request_by_employee_id(id)
sp_select_event_request_by_id(id)
sp_update_request_type(old event request, new event request)
sp_deactivate_event_request(id)
sp_delete_event_request(id)
sp_authenticate_event_type(event type)
sp_authenticate_employee(id)
```

```
sp_insert_event(event)
sp_select_all_events()
sp_select_event_by_id(id)
sp_select_event_by_type(type)
```

sp\_update\_event(old event, new event)

sp\_deactivate\_event(id)

sp\_delete\_event(id)

sp\_verify\_event\_request()

sp\_insert\_item\_order\_line(order line)

sp\_retrieve\_line\_by\_item\_order\_id(id)

sp\_retrieve\_order\_total(id)

sp\_retrieve\_order\_line\_count(id)

sp\_update\_order\_qty(Qty)

sp\_update\_supplier\_id(id)

sp\_delete\_line(id)

sp\_insert\_pet(id)

sp\_retrieve\_all\_pets(id)

sp\_update\_pet(id)

sp\_delete\_pet(PetId)

sp\_insert\_pet\_type(int petID, string petType)

sp\_update\_pet\_type(int petID, string petType)

sp\_delete\_pet\_type(int petID)

sp\_deactivate\_pet\_type(int petID)

sp\_select\_pet\_type(int petID)

sp\_select\_all\_pet\_types()

sp\_insert\_vehicle(id)

sp\_retrieve\_all\_vehicles

sp\_retrieve\_vehicle\_by\_id(id)

sp\_update\_vehicle(id)

```
sp_update_vehicle(int)
sp_delete_vehicle(int)

sp_insert_item_supplier(int)
sp_select_item_order_by_id(int)
sp_update_item_supplier(int)
sp_delete_item_supplier(int)
sp_select_item_orders()
sp_deactivate_item_supplier(int)

sp_insert_item_supplier(NVarchar(17))
sp_select_item_supplier_by_id(NVarchar(17))
sp_update_item_supplier(NVarchar(17))
sp_delete_item_supplier(NVarchar(17))
sp_select_item_suppliers()
sp_deactivate_item_supplier(NVarchar(17))

sp_insert_supplier(int)
sp_select_supplier_by_id(int)
sp_update_supplier(int)
sp_delete_supplier(int)
sp_select_suppliers()
sp_deactivate_supplier(int)

sp_insert_supplier_order(int orderID, int supplierID, DateTime editDate)
sp_select_all_supplier_orders()
sp_select_supplier_order(int orderID, int supplierID)
sp_update_supplier_order(int orderID, int supplierID, DateTime editDate)
sp_delete_supplier_order(int orderID, int supplierID)
```

```
sp_deactivate_supplier_order(int orderID, int supplierID)

sp_insert_appointment_type(int appointmentID, type nvarchar())
sp_update_appointment_type((int appointmentID, type nvarchar()))
sp_select_all_appointment_types()
sp_select_appointment_type(int appointmentID)
sp_delete_appointment_type(int appointmentID)
sp_deactivate_appointment_type(int appointmentID)

sp_insert_shop(int shopID, string shop)
sp_update_shop(int shop, string shop)
sp_delete_shop(int shopID)
sp_deactivate_shop(int shopID)
sp_select_shop(int shopID)
sp_select_all_shops()
```

## Contributors

Caitlin Abelson

Ramesh Adhikari

Carlos Arzu

Craig Barkley

Austin Berquam

Eric Bostwick

Kevin Broskow

Richard Carroll

Dalton Cleveland

Eduardo Colon

Austin Delaney

Jared Greenfield

Benjamin Hanna

Phillip Hansen

James Heim

Cody Herb

Matthew Hill

Matt LaMarche

Jacob Miller

Miyala Mingomba

Wesley Richardson

Alisa Roehr

Dani Russo

Gunardi Saputra

Chase Schulte

Jeffrey Stobb

Jesse Tomash

