Software Design Descriptions

Tiffany Wise

University of Maryland University College

CMIS 330 – Section 7980

Professor Abdulnasir Shash

Table of Contents

1. Introductions

1.1. Purpose

This document provides a detailed description of the software design of the reservation and accounting architecture, data, procedural, and interface designs used in the bed and breakfast software system.

1.2. Definitions and acronyms

Data entity – Element of a software system, such as an object, field, function, or component

Attribute – descriptor (i.e. field) of an object

2. References

IEEE Standard 1016 - 1998

3. Architectural Design

3.1. Business Architecture

The business architecture shows the overlap between the software system and the various hotel business processes, employees, and customers. It provides insight into the frequently-used portions of the software, which can be used to determine system maintenance needs and future process improvements. The business architecture is shown in Figure 1.

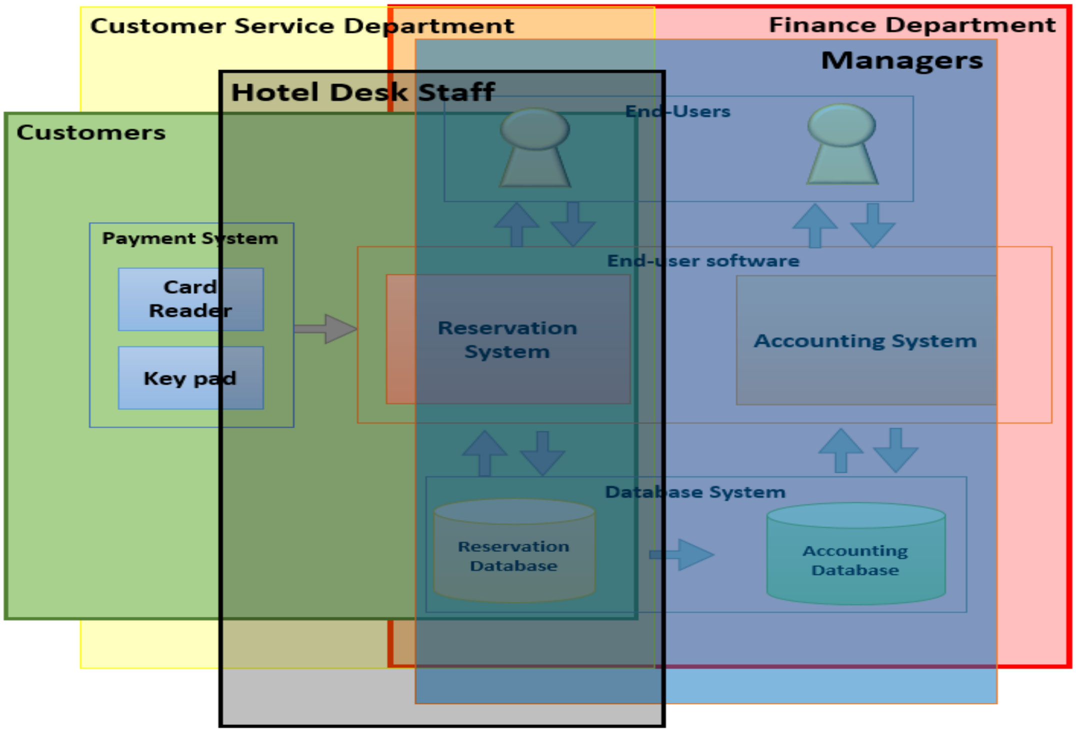


Figure 1. Architectural Context Diagram of Business Architecture

The reservation software subsystem, which consists of the reservation software and database, is used in all aspects of the business. The accounting and payment subsystems interact with a portion of the business processes and end-users.

3.2. System Architecture

The end-user software contains the reservation and accounting system software packages that operate the entire system. The payment system uses allows the hotel to accept payments via credit or debit card. The database system contains databases for reservation and accounting data. End-users control the system through interfaces for the reservation and accounting software. Figure 2 shows the interaction between subsystems and the flow of data between them.

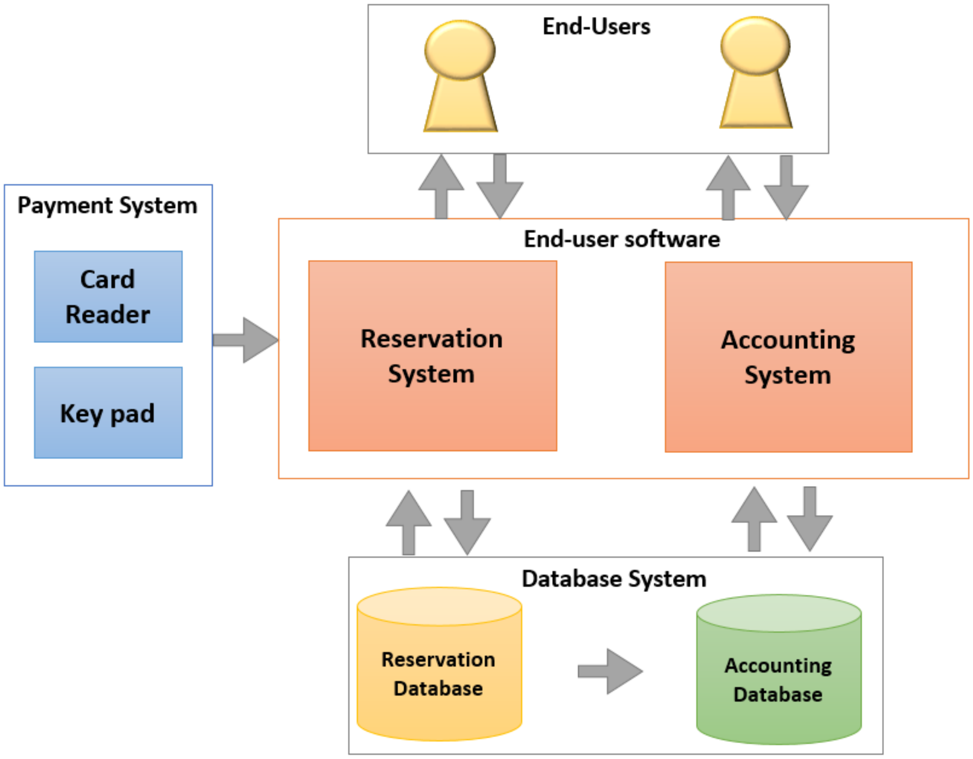


Figure 2. Data Flow Diagram (Level 0)

3.3. Class Architecture

The class architecture for the end-user software consists a single Reservation class and four abstract classes for rooms, persons, dates, and financial records.

- Reservation System. The reservation system contains the Reservation class which contains subclass objects of the abstract classes: Room, Person, and HotelDate.

- Accounting System. The Accounting system contains a single abstract class, FinancialRecord, with subclasses Income and Expense. It also contains Person subclass objects: Payee and Payer.

The class architecture is show in Figure 3. A more detailed class diagram will be provided.

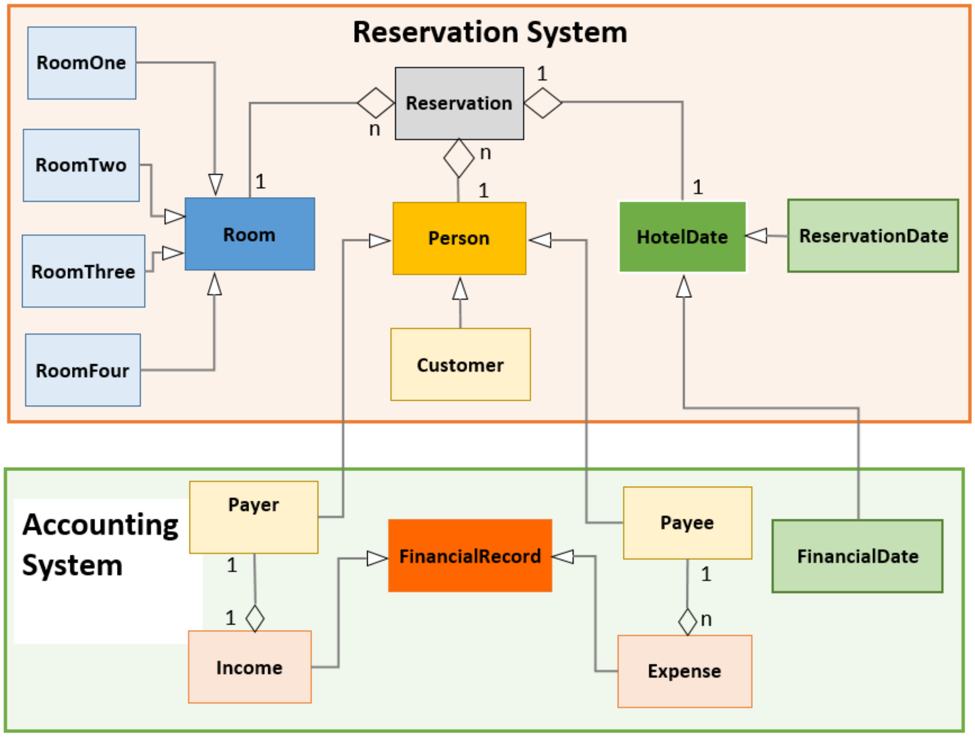


Figure 3. Class architecture

3.4. Xx

4. Data Design

4.1. Data Description

The following class diagram, Figure 1, shows all of the entities used in the Reservation and Accounting subsystems as well as their attributes and methods.

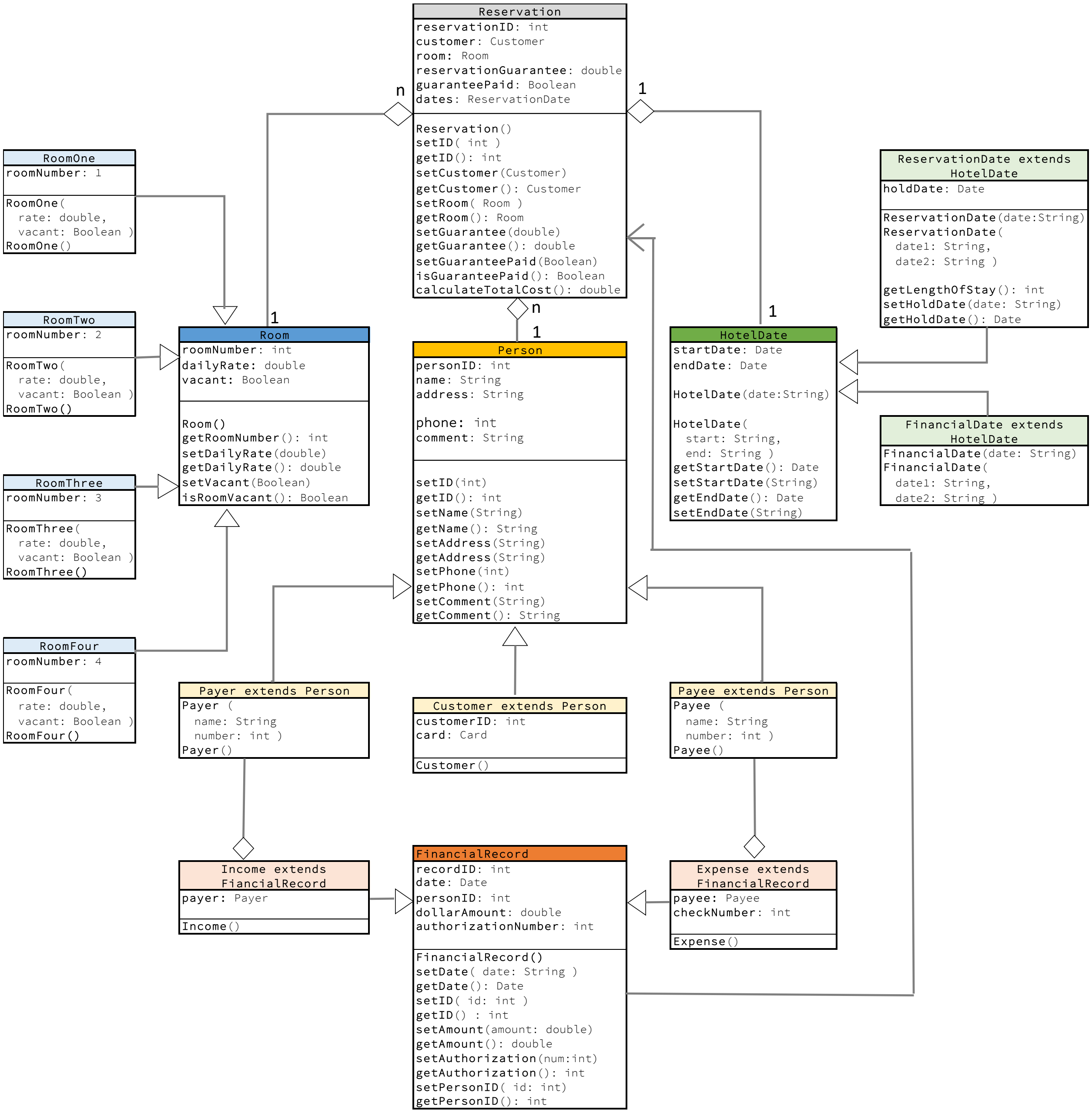


Figure 4. Class diagram of Reservation and Accounting System

4.2. Data Dictionary

The data dictionary will contain an explanation of the purpose of each class, its attributes and methods, and how each interacts with other data entities.

4.2.1. Reservation Class

Purpose. The Reservation Class holds all pertinent information regarding a hotel reservation. It contains information on the room, customer, payment information, and dates of reservation.

Function. It creates a hotel reservation.

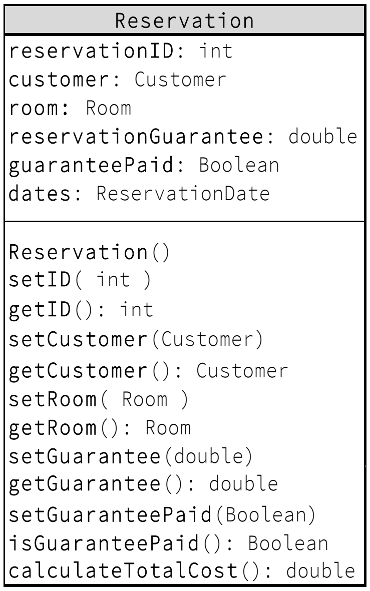


Image 1. Reservation class

Attributes

Reservation ID number

Variable Name	reservationID
Variable Type	int
Description	Unique number that identifies reservation in the database
Interactions	Assigned by the database after a room is selected and customer information is entered into database

Customer Object

Variable Name	customer
Variable Type	Customer object
Description	Contains customer ID, name, address, phone number, credit card
Interactions	Created when customer information is entered during reservation.

Reserved Room.

Variable Name	room
Variable Type	Child object of Room class
Description	Contains room number, daily rate, vacancy status
Interactions	Created when room is selected after searching for vacancy

Reservation Guarantee.

Variable Name	reservationGuarantee
Variable Type	double
Description	Price to reserve a room at the bed and breakfast

Interactions	Retrieved from Reservation database when room is vacant during dates of reservation

Reservation Guarantee Confirmation.

Variable Name	guaranteePaid
Variable Type	Boolean
Description	Indicates if the reservation guarantee has been paid
Interactions	Retrieved from Reservation Database after Reservation is created. Used to hold a reservation with or without the reservation guarantee being paid.

Reservation Dates.

Variable Name	dates
Variable Type	ReservationDate object
Description	Start and ending date of Reservation
Interactions	Created during initial search for vacancies. Added to Reservation object after Customer selects Room.

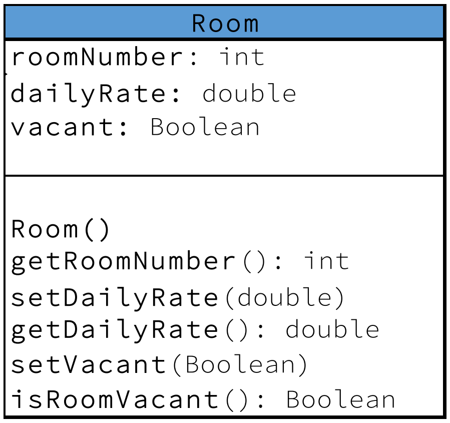
Methods

- **setID** and **getID** – Assigns and returns reservation ID number
- **setCustomer** and **getCustomer** – Assigns and returns Customer object
- **setRoom** and **getRoom** – Assigns and returns Room object
- **setGuarantee** and **getGuarantee** – Assigns and returns Reservation Guarantee price
- **setGuaranteePaid** and **isGuaranteePaid** – Sets and returns Reservation Guarantee Paid status
- **calculateTotalCost** – determines total cost for reservation

4.2.2. Room class

Purpose. The Room class is an abstract class that contains attributes and functions shared by all rooms in the bed and breakfast. It allows for the addition of more rooms, if needed.

Function. It stores the room number, daily price, and vacancy status of a given room.



Attributes.

Room Number

Variable Name	room
Variable Type	Int
Description	Assigned room number
Interactions	Interacts with subclasses: RoomOne, RoomTwo, RoomThree, RoomFour

Daily Rate

Variable Name	dailyRate
Variable Type	Double
Description	Price per day to reserve a specific room
Interactions	Retrieved from database during vacancy search. Added to Reservation object as Reservation Guarantee price after Customer selects Room and customer information (i.e. Customer object created) is entered.

Vacancy Status

Variable Name	vacant
Variable Type	Boolean
Description	Indicates if a room is vacant or reserved
Interactions	Retrieved from Reservation database during search

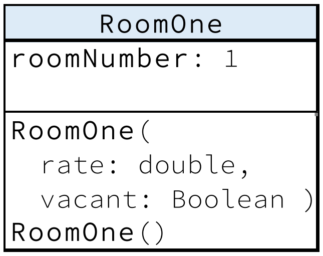
Methods

- **getRoomNumber** – returns Room's number
- **setDailyRate** and **getDailyRate** – sets and returns the daily price of a Room
- **setVacant** and **isRooomVacant** – sets and returns Room vacancy status

4.2.2.1. RoomOne

Purpose. Store information on RoomOne about its daily price and vacancy status.

Function. Edit and retrieve reservation pricing and vacancy status

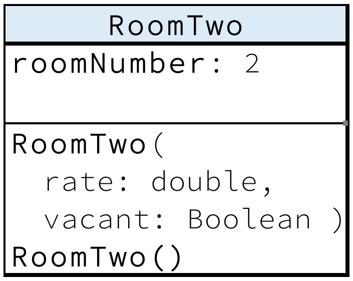


Attributes.

Room Number

Variable Name	roomNumber = 1
Variable Type	static Int
Description	Assigned room number
Interactions	Retrieves price and vacancy status from Reservation database. Added to Reservation object before Customer Information is added.

4.2.2.2. RoomTwo



Purpose. Store information on RoomTwo about its daily price and vacancy status.

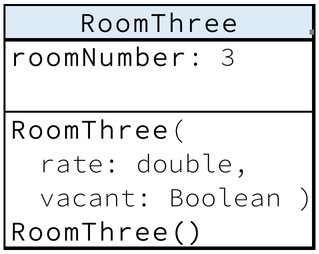
Function. Edit and retrieve reservation pricing and vacancy status

Attributes.

Room Number

Variable Name	roomNumber = 2
Variable Type	static Int
Description	Assigned room number
Interactions	Retrieves price and vacancy status from Reservation database. Added to Reservation object before Customer Information is added.

4.2.2.3. RoomThree



Purpose. Store information on RoomThree about its daily price and vacancy status.

Function. Edit and retrieve reservation pricing and vacancy status

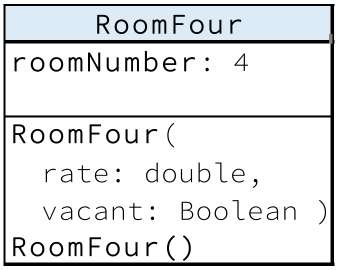
Attributes.

Room Number

Variable Name	roomNumber = 3
Variable Type	static Int
Description	Assigned room number

Interactions	Retrieves price and vacancy status from Reservation database. Added to Reservation object before Customer Information is added.

4.2.2.4. RoomFour



Purpose. Store information on Roomfour about its daily price and vacancy status.

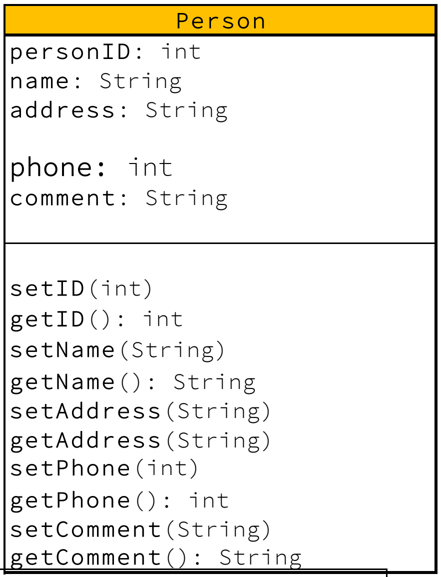
Function. Edit and retrieve reservation pricing and vacancy status

Attributes.

Room Number

Variable Name	roomNumber = 4
Variable Type	static Int
Description	Assigned room number
Interactions	Retrieves price and vacancy status from Reservation database. Added to Reservation object before Customer Information is added.

4.2.3. Person class



Purpose. The Person class is an abstract class that stores common attributes and methods for its subclasses: Customer, Payer, Payee.

Function. It stores the name, address, and phone numbers for customers, payers, and payees.

Attributes

Person ID number

Variable Name	personID
Variable Type	int
Description	Unique identifier for a Person object
Interactions	Assigned by Reservation or Accounting database

Name

Variable Name	name
Variable Type	String
Description	First and last name of Person
Interactions	Entered by end -user in Reservation or Accounting interface

Address

Variable Name	address
Variable Type	String
Description	Address of Person
Interactions	Entered by end -user in Reservation or Accounting interface

Phone

Variable Name	phone
Variable Type	int
Description	Phone number of Person
Interactions	Entered by end -user in Reservation or Accounting interface

Comment

Variable Name	comment
Variable Type	String
Description	Comment regarding a Person
Interactions	Entered by end -user in Accounting interface

Methods

- **setID** or **getID** – set or return Person ID numbers
- **setName** or **getName** – set or return Person name
- **setAddress** or **getAddress** – set or return Person address
- **setPhone** or **getPhone** – set or return Person phone number
- **setComment** or **getComment** – set or return Person comment

4.2.3.1. Customer

Purpose. Stores information about reservation customers.

Function. Provides a Customer's name, address, phone number, and payment card information

Attributes.

Customer ID number

Variable Name	customerID
Variable Type	int
Description	Unique identifier for Customer
Interactions	Assigned by Reservation database after customer information is saved during vacancy search

Card

Variable Name	card
Variable Type	Card object
Description	Card number, PIN number, payment verification, payment confirmation number
Interactions	Saved when processing payment for a reservation. Interacts with Reservation database through Reservation object. Interacts with Accounting database through Reservation database.

4.2.3.2. Payer

Purpose. Stores information about Payer

Function. Stores Payer's name, address, ID number, and comment.

Attributes.

Payer ID number

Variable Name	payerID	
Variable Type	int	
Description	Unique identifier for Payer	
Interactions	Assigned by Accounting database after Payer information is saved during financial record data entry	

Methods. No additional methods. See Person parent class.

4.2.3.3. Payee

Purpose. Stores information about Payee

Function. Stores Payee's name, address, ID number, and comment.

Attributes.

Payee ID number

Variable Name	payeeID
Variable Type	int
Description	Unique identifier for Payee
Interactions	Assigned by Accounting database after Payee information is saved during financial record data entry

Methods. No additional methods. See Person parent class.

4.2.4. HotelDate class

```
HotelDate
startDate: Date
endDate: Date

HotelDate(date:String)

HotelDate(
  start: String,
  end: String )
getStartDate(): Date
setStartDate(String)
getEndDate(): Date
setEndDate(String)
```

Purpose. Abstract class that stores starting and ending dates used in reservation and financial record searches

Function. Provides information on the starting and ending dates of a reservation or financial search

Attributes.

Start Date

Variable Name	startDate
Variable Type	Date
Description	Starting date of a reservation or financial record search
Interactions	Created when search is initiated after date validation. Interacts with Reservation objects and FiancialRecord objects.

End Date

Variable Name	endDate
Variable Type	Date
Description	Ending date of a reservation or financial record search
Interactions	Created when search is initiated after date validation. Interacts with Reservation objects and FiancialRecord objects.

Methods

- **getStartDate** and **setStartDate** – sets and returns start dates
- **getEndDate** and **setEndDate** – sets and returns ending dates

4.2.4.1. ReservationDate

Purpose. Stores starting and ending dates of a hotel reservation

Function. Provides information about length of reservation, reservation hold date, and starting and ending dates

Attributes.

Hold Date

```
ReservationDate extends
      HotelDate
------------------------------
holdDate: Date

------------------------------
ReservationDate(date:String)
ReservationDate(
  date1: String,
  date2: String )

getLengthOfStay(): int
setHoldDate(date: String)
getHoldDate(): Date
```

Variable Name	holdDate
Variable Type	Date object
Description	The last date that a reservation will be held. This date is specified during a Hold Reservation when the customer does not pay the reservation guarantee fee. The date is set to the ending date of a Confirmed Reservation when the customer pays the reservation guarantee fee.
Interactions	Interacts with Reservation objects and Reservation database

Methods

- **getLengthOfStay** – calculates the number of days between starting and end dates of reservation
- **setHoldDate** and **getHoldDate**– sets and returns the Reservation Hold date

4.2.4.2. FinancialDate

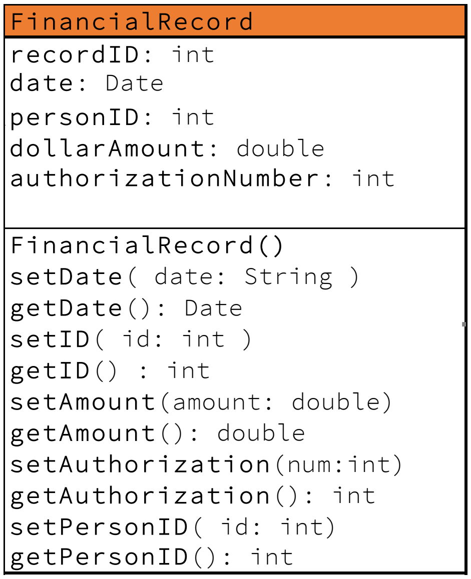
Purpose. Stores information about the starting and end dates of a financial record search.

Function. Used to search Accounting database

Attributes. No additional attributes added. See HotelDate parent class.

Methods. No additional methods added. See HotelDate parent class.

4.2.5. FinancialRecord class



```
FinancialRecord
recordID: int
date: Date
personID: int
dollarAmount: double
authorizationNumber: int

FinancialRecord()
setDate( date: String )
getDate(): Date
setID( id: int )
getID() : int
setAmount(amount: double)
getAmount(): double
setAuthorization(num:int)
getAuthorization(): int
setPersonID( id: int)
getPersonID(): int
```

Purpose. Abstract class that is used to store information regarding the income or expense of a financial record, such as record ID number, date of occurrence, Payer/Payee ID number, dollar amount, and authorization number.

Function. Financial data is added to a FinancialRecord object, which will be added to an Income or Expense object once the end-user indicates the type of financial record is being created.

Attributes.

Record ID number

Variable Name	recordID
Variable Type	Int
Description	Unique identifier for financial record
Interactions	Assigned by Accounting database

Date of Occurrence

Variable Name	date
Variable Type	Date object
Description	Indicates when income or expense occurred
Interactions	Interacts with Income and Expense objects

Person ID number

Variable Name	personID
Variable Type	Int
Description	Unique identifier for Person
Interactions	Assigned by Accounting database

Dollar amount

Variable Name	dollarAmount
Variable Type	double
Description	Amount of money received or paid

Interactions	Entered by end-user in Accounting end-user interface or from Reservation object in Reservation database

Authorization number

Variable Name	authorizationNumber
Variable Type	Int
Description	Unique identifier Person
Interactions	Entered by end-user in Accounting interface

Methods

- **setDate** and **endDate** – sets and returns date of occurrence
- **setID** and **getID –** sets and returns record ID number
- **setAmount** and **getAmount** – sets and returns dollar amount
- **setAuthorization** and **getAuthorization –** sets and returns authorization number
- **setPersonID** and **getPersonID** - sets and returns person ID number

4.2.5.1. Income

Purpose. Stores information about hotel income

Function. Proves information about when income is received, the amount, and from where.

Attributes.

Payer

Variable Name	payer
Variable Type	Payer object
Description	Person who provided income
Interactions	Entered by end-user in Accounting interface

Methods. No additional methods. See Person parent class.

4.2.5.2. Expense

Purpose. Stores information about hotel expense

Function. Proves information about when expenses are paid, the amount, and to whom.

Attributes.

Payee

Variable Name	payee
Variable Type	Payee object
Description	Person who received payment
Interactions	Entered by end-user in Accounting interface

Check Number

Variable Name	checkNumber
Variable Type	int
Description	Check number on check used to pay Payee
Interactions	Provided by Accounting database

Methods. No additional methods. See Person parent class.

5. Procedural Design for the Reservation System

5.1. Data Flow Diagram (Level 1) and Process Description

The procedural design describes the specific processes that occur throughout various states of operation in the software. The overall processes that occur in the Reservation system are shown in Figure 5. A description of the processes are below the image.

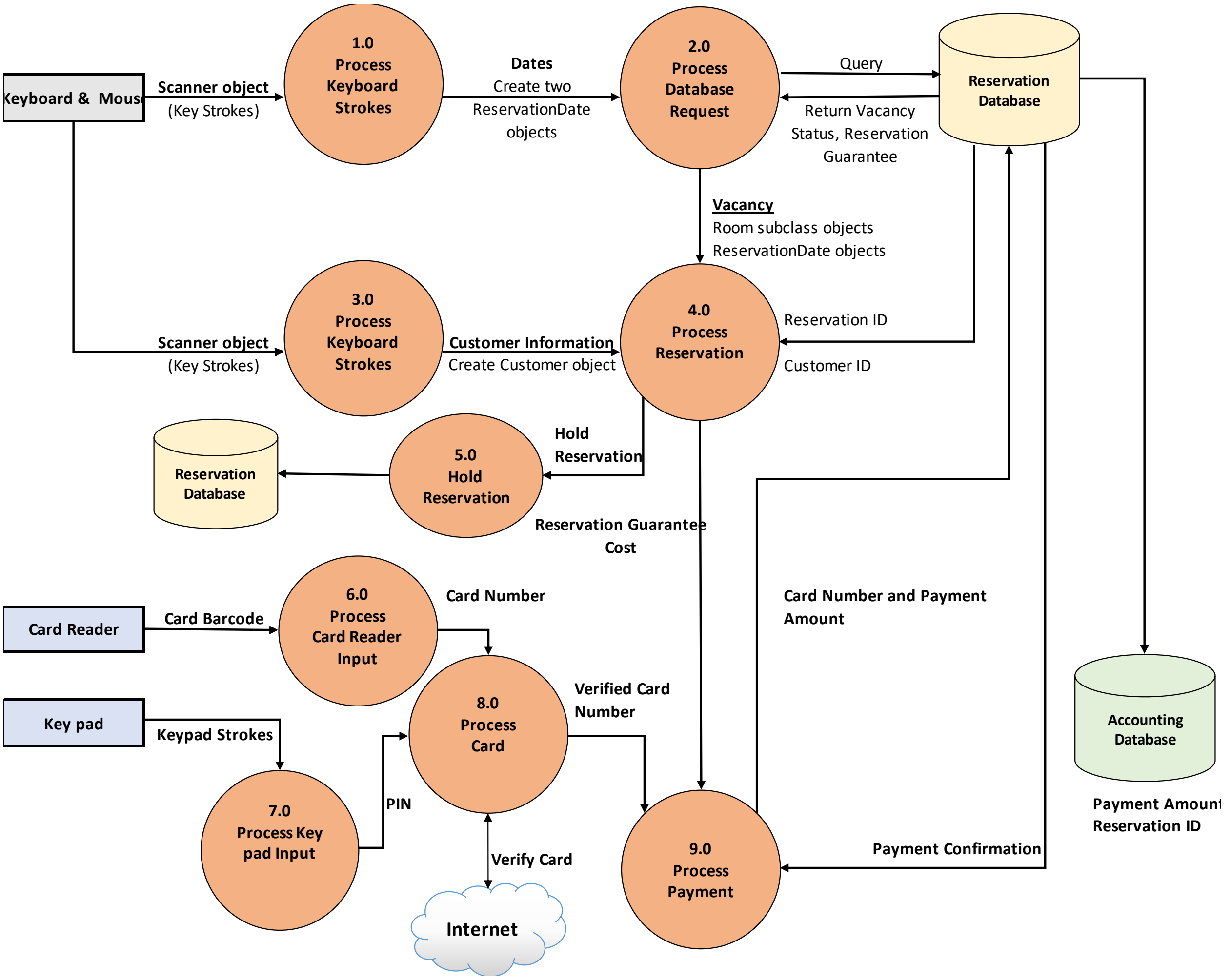


Figure 5. Data Flow Diagram (Level 1)

Processes

Process 1.0 Process Key Strokes
Reservation dates are entered into Reservation end-user interface. The data is converted into ReservationDate objects and sent to the database.

Process 2.0 Process Database Request
Reservation dates are sent to Reservation database. The database returns information on the rooms that are vacant during the selected dates and the daily prices of those rooms.

Process 3.0 Process Key Strokes
Customer chooses a room (Room object). Then, end-user enters customer information is entered into Reservation end-user interface. A Customer object is created.

Process 4.0 Process Reservation
The customer information (Customer object), room information (Room object), and reservation dates (ReservationDate object) are used to create begin a reservation (Reservation object). The reservation information (Reservation object) is sent to the Reservation database, which returns Reservation and Customer ID numbers. The reservation information (Reservation object) is also sent to Hold the Reservation (Process 5.0) or Process Payment (Process 9.0).

Process 5.0 Hold Reservation
The customer chooses not to pay the Reservation Guarantee price. The reservation is held until a certain date, entered by the end-user. The reservation hold is sent to the Reservation database with a status of awaiting payment.

Process 6.0 Process Card Reader Input
The customer pays the Reservation Guarantee price with their credit or debit card. The card number is validated before it is sent to the verification process (Process 8.0).

Process 7.0 Process Key Pad Input
If the customer pays with a debit card, a PIN number is entered. The PIN number is validated before it is sent to the verification process (Process 8.0).

Process 8.0 Process Card
The debit card and PIN number or the credit card are verified. The verified card information is sent for payment processing (Process 9.0).

Process 9.0 Process Payment
The reservation guarantee price (Reservation object) and card information (Card object) are used to charge the card. The card information (Card object) and payment amount are sent to the Reservation database, which returns a payment confirmation number. The Reservation database sends the payment amount and reservation ID to the Accounting database.

5.2. Data Flow Diagram (Level 2) and Process Description

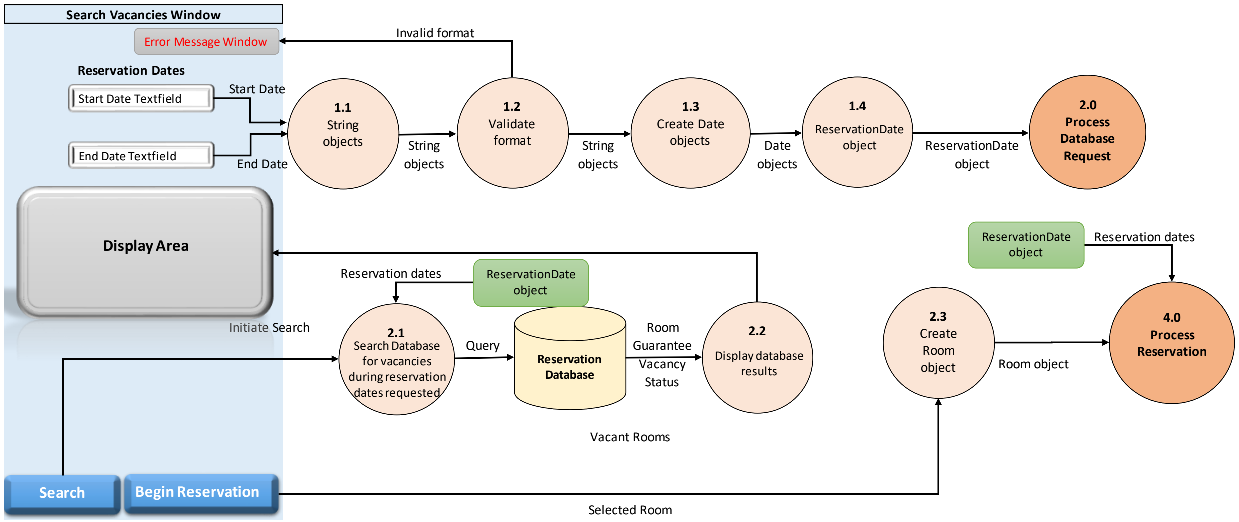


Figure 6. Data Flow Diagram (Level 2) for Processes 1.0 and 2.0. In this scenario, the end-user is searching for vacant rooms on a given date. A reservation is initiated from selected vacancy.

5.2.1. Searching for a Vacancy (Process 1.0 Process Key Strokes)

The end-user enters reservation dates, which are validated. Invalid dates display an error message in the Error Message Window. The valid dates are converted into Date objects, which are added to a ReservationDate object.

5.2.2. Viewing Search Results and Beginning a Reservation (Process 2.0 Process Database Request)

After the end-user clicks the Search button, reservation dates from the ReservationDate object are sent to the Reservation database. The database sends back available rooms and their daily price (Reservation Guarantee). The vacant rooms and prices are displayed on the Display Area of the Reservation system end-user interface.

The customer choose a room. The end-user selects the room and clicks Begin Reservation. A Room object is created and is ready for reservation processing.

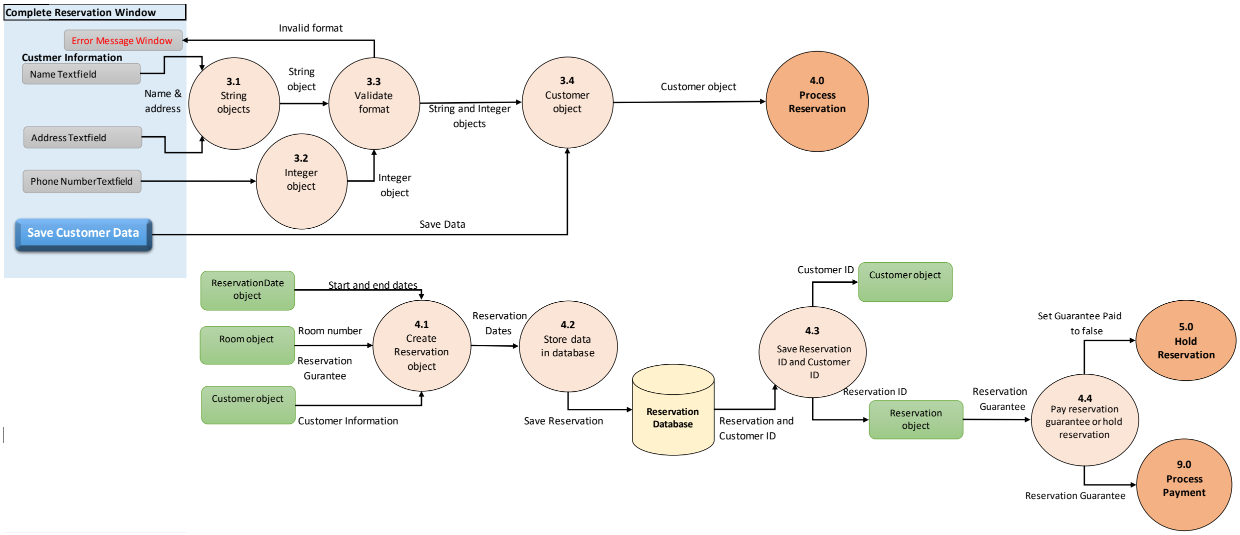


Figure 7. Data Flow Diagram (Level 2) for Processes 3.0 and 4.0. The end-user enters customer information, which creates a Reservation object. The customer can pay to hold the reservation now or pay by a certain date.

5.2.3. Enter Customer Information (**Process 3.0** Process Key Strokes)

The end-user enters the customer's name, address, and phone number into the end-user interface and clicks the Save Customer Data button. The information is validated as the end-user types into the text fields and an error message appears for any invalid input. Valid input is saved as a Customer object.

5.2.4. Create a Reservation (**Process 4.0** Process Reservation)

The reservation dates (ReservationDate object), room information (Room object), and customer information (Customer object) are added to a Reservation object. The Reservation object sends the reservation dates to the Reservation database, which returns a reservation ID number and a customer ID number, which are added to the Reservation and Customer objects, respectively. The customer can choose to pay the hold reservation fee (Reservation Guarantee, Process 9.0) now or at a later date (Process 5.0).

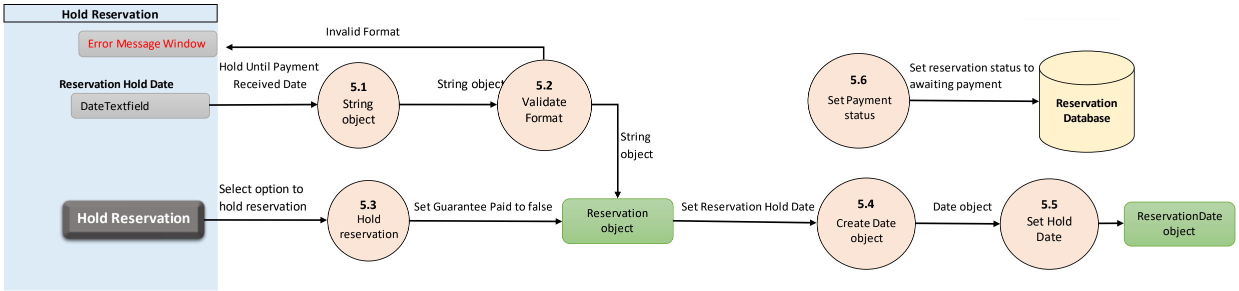


Figure 8. Holding a Reservation (Process 5.0). The customer chooses to postpone payment for a reservation

5.2.5. Holding a Reservation (**Process 5.0** Hold Reservation)

The customer requests to pay for reservation confirmation at a later date. The end-user enters the chosen date in the Reservation Hold Date text field. The date is validated; Invalid data generates an error message in the Error Message Window. The hold date is added to the Reservation object after the end-user selects the Hold Reservation button.

The Reservation object creates a Date object, which it adds to the ReservationDate object’s holdDate attribute field. Reservation guarantee payment status is set to awaiting payment in the Reservation database.

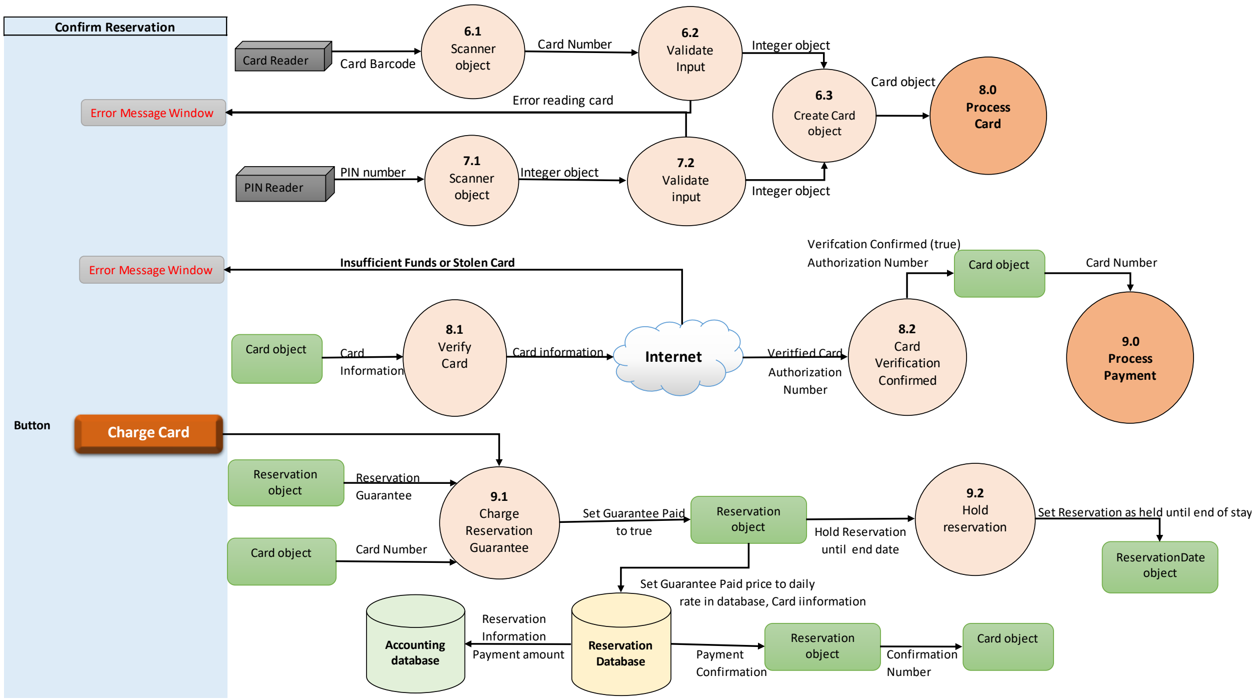


Figure 9. Data Flow Diagram (Level 2) for Processes 6 through 9. Confirming a Reservation.

5.2.6. Customer swipes Credit Card (**Process 6.0** Process Card Reader Input)

The customer swipes their credit card or debit card into the card reader. The card number is validated. An invalid card number generates a message on the Error Message Window. The valid number is added to a Card object.

5.2.7. Customer enters PIN number (**Process 7.0** Process Key Pad Input)

If the customer uses a debit card, the customer will enter a PIN number into the key pad reader. The PIN number is validated. An invalid PIN number generates a message on the Error Message Window. The valid number is added to the Card object.

5.2.8. Card Verification and Authorization (**Process 8.0** Process Card)

Card information is verified. Information regarding Insufficient funds or a reported stolen credit card appears on the Error Message Window. Verification Confirmation is sent to the Card object. An authorization number will be used for payment processing.

5.2.9. **Process 9.0** Process Payment

The end-user presses the Charge Card button. The card number (Card object) and reservation guarantee fee (Reservation object) are used to charge the card. Reservation Guarantee status is set to true in the Reservation object, which adds the ending date of the reservation as the Reservation Hold date in the ReservationDate object.

The Reservation object sends reservation and card information to the Reservation database. The Reservation database sends a confirmation number to the Card object, via the Reservation object. The database also sends customer information, payment information, and reservation ID to the Accounting database.

6. Procedural Design for the Accounting System

6.1. Data Flow Diagram (Level 1) and Process Description

The overall processes that occur in the Accounting System are shown in Figure 10.

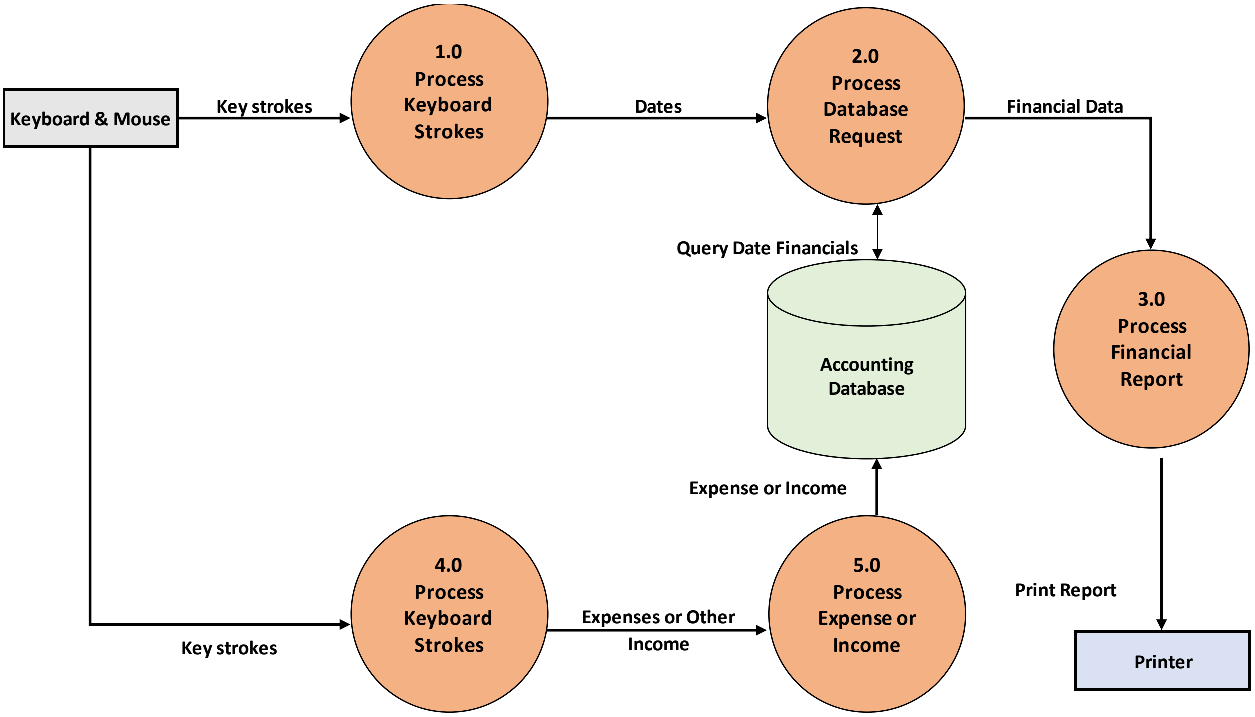


Figure 10.

Process 1.0 Process Keyboard Strokes
The end-user enters start and end dates to search for financial records. The resulting Date objects are sent to query the database.

Process 2.0 Process Database Request
The Accounting database returns all financial records that occurred on the entered dates.

Process 3.0 Process Financial Report
A financial report is created and can be printed.

Process 4.0 Process Keyboard Strokes
The end-user enters information about a hotel income or expense.

Process 5.0 Process Expense or Income
Based on user input, an Expense object or Income object is created and added to the Accounting database.

6.2. Data Flow Diagram (Level 2) and Process Description

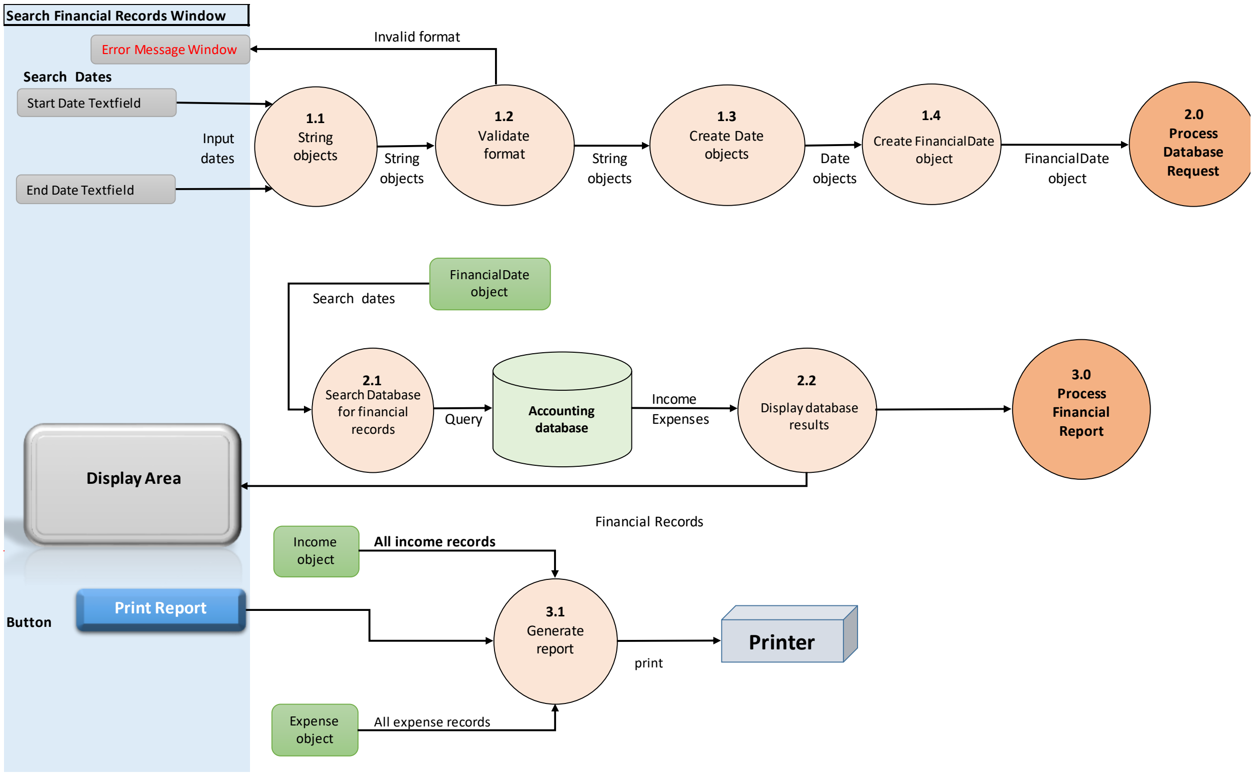


Figure 11. Data Flow Diagram (Level 2) for Processes 1.0 to 3.0. The end-user searches for financial records and prints a report.

6.2.1. Enter starting and ending dates for search (Process 1.0 Process Keyboard Strokes)

Start and end dates are validated. Invalid data generates error message in Error Message Window. Valid data generates Date objects, which are added to a FinancialDate object.

6.2.2. Query Accounting database and view results (Process 2.0 Process database request)

Dates from FinancialDate object are used to query Accounting database. The database returns all financial records and displays them on end-user Accounting interface.

6.2.3. Print Financial report (Process 2.0 Process Financial report)

End-user presses Print Report button. All income (Income objects) and expenses (Expense objects) print.

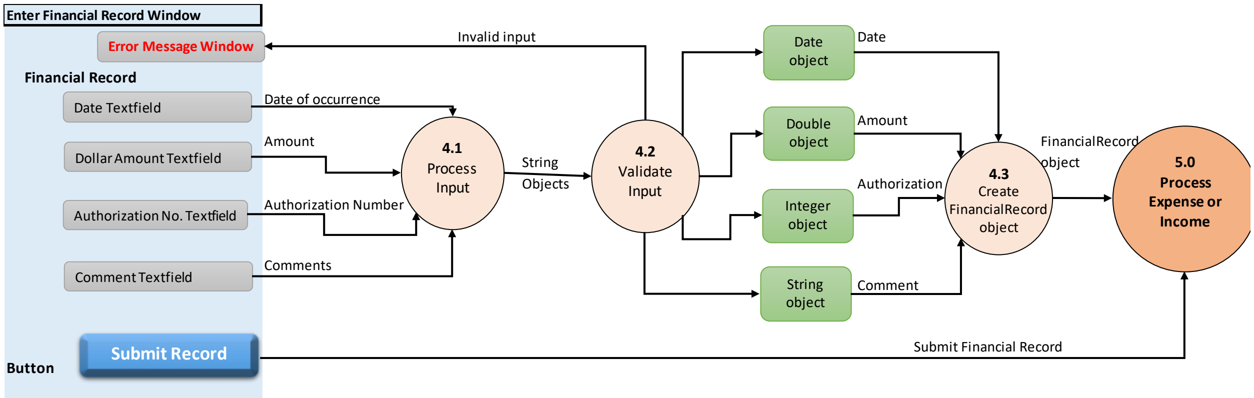


Figure 12. Data Flow Diagram (Level 2) from Process 4.0. End-user enters information about a new financial record.

6.2.4. Enter financial data (Process 4.0 Process Keyboard Strokes)

End-user eners date, dollar amount, authorization number, and any comment into text fields. Invalid data generates error message in Error Message Window. Valid data is created into Date objects, Double objects, Integer objects, and String objects, which are added to a FinancialRecord object.

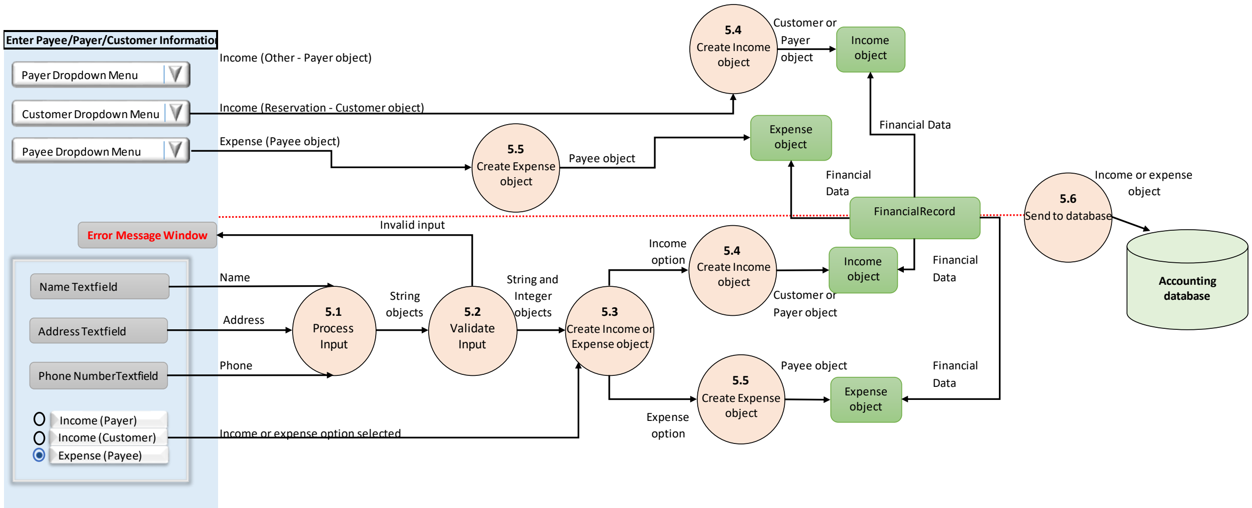


Figure 13. Data Flow Diagram (Level 2) from Process 5.0. End-user selects from previous Payers, Payees, or Customers, or manually adds information.

6.2.5. Enter information about Person receiving or sending payment (Process 5.0 Process Expense or Income)

If the End-user selects from the a drop-down menu, a Payee object is created for a Payee drop-down menu selection or a Payer object is created for a Payer drop-down menu selection.

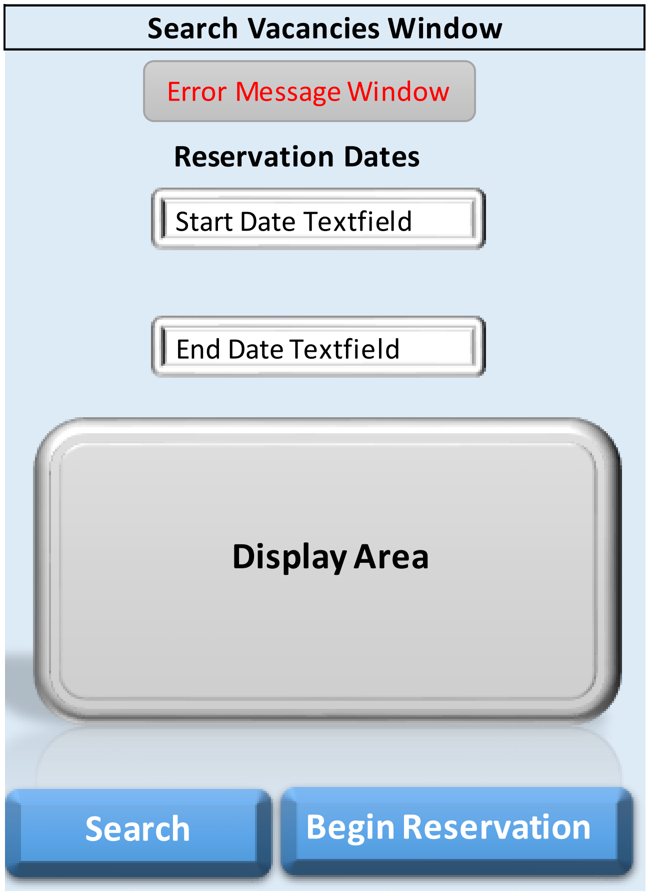
An Income object is created to add the Payer object information to it. An Expense object is created to add a Payee object to it. Financial information from the FinancialRecord is added to the Income or Expense object.

If the end-user manually enters Payer or Payee information, the information is validated and the end-user must select an option from the radio button group. The radio button selection creates the appropriate objects.

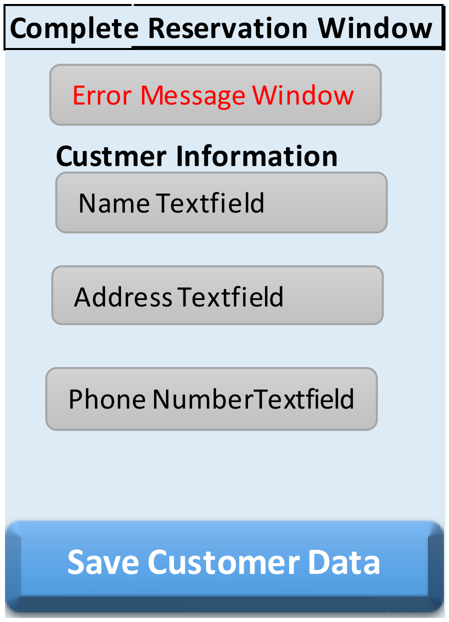
7. Interface Design

7.1. Human Computer Interface for Reservation System

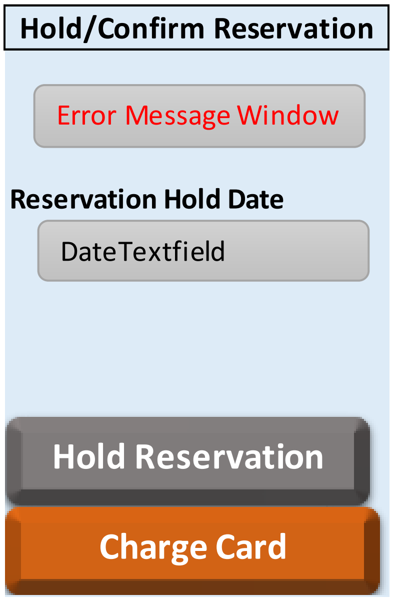
7.1.1. Search for Reservation Vacancies



7.1.2. Complete Reservation

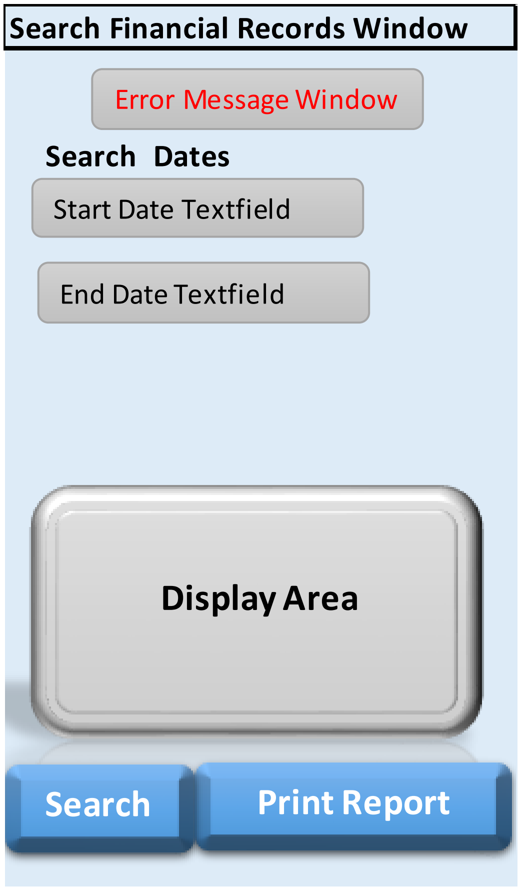


7.1.3. Hold or Confirm Reservation

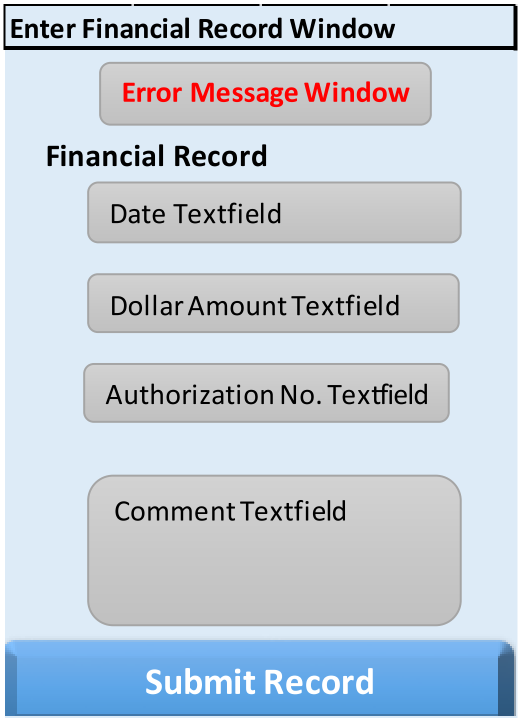


7.2. Human Computer Interface for Accounting System

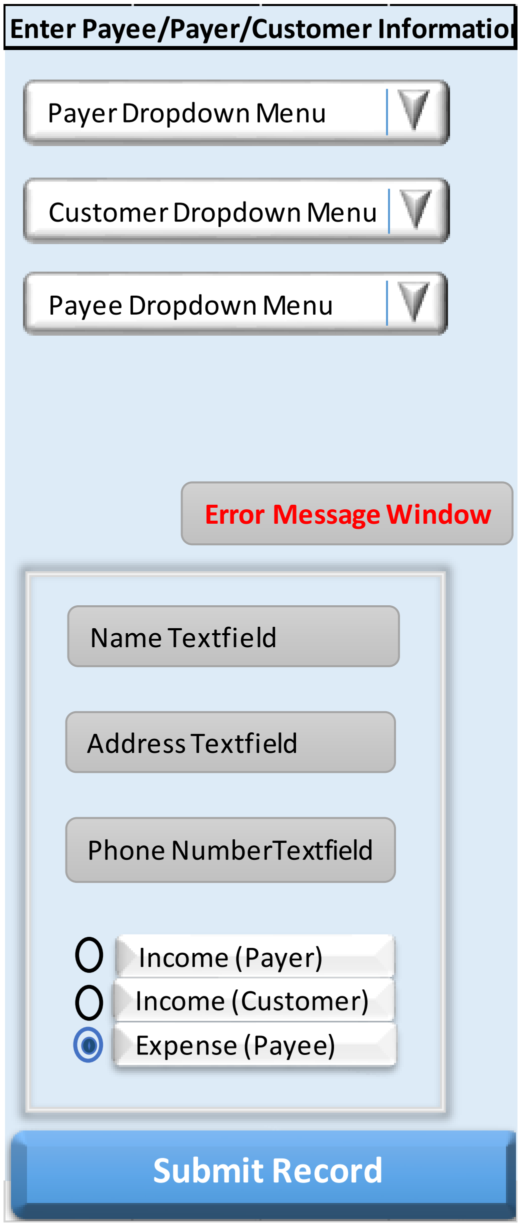
7.2.1. Search for Financial Records



7.2.2. Enter New Financial Record



7.2.3. Enter Payee or Payer Information



7.3. Software System Interface

The Reservation and Accounting software subsystems communicate with each other through the Reservation and Accounting Databases. This interface is controlled by database software.

7.4. Software and External Entity Interface

7.4.1. Reservation System and Payment System

The Key Pad Reader and Card Reader will plug into the ports on the computer where the Reservation system is installed. The Reservation software will read the input from these ports.

7.4.2. Accounting System and Printer

The Printer will plug into the computer’s USB port. The Accounting system will send financial reports through the USB port for printing.