Michael Rodriguez **CMIS 330/6380** **Professor Morad, Osama**

John and Jane's Bed & Breakfast Software Requirements Sheet

1. Introduction

1.1 Purpose - The purpose of this Software Requirements Sheet is to document the requirements required to develop a software product capable of handling the needs of the John and Jane's Bed & Breakfast.

1.2 Scope – This Software Requirements Sheet shall cover the desktop application required to aid in the management of the John and Jane Bed & Breakfast.

1.3 Definitions, Acronyms, and Abbreviations – From here on The Software Requirements Sheet will be referred as SRS and John and Jane's Bed & Breakfast as B&B. GUI is the Graphical User Interface that the B&B staff will be using within the system. The Card Verification Code will be referred to as "CVC".

1.4 References-

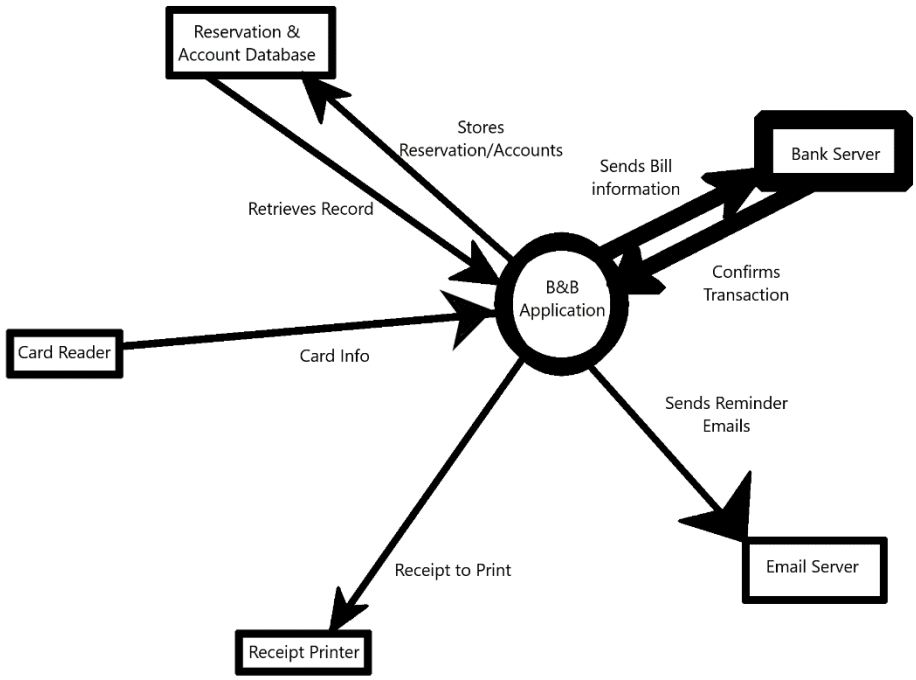
Statement of Need: John and Jane are starting a bed-and-breakfast (B&B) in a small New England town. They will have four bedrooms for guests. They want a system to manage the reservations and to monitor expenses and profits. When a potential customer calls for a reservation, they will check the calendar, and if there is a vacancy, they will enter the customer's name, address, and phone number, dates, agreed upon price, credit card number, and room numbers. Reservations must be guaranteed by 1 day's payment. Reservations will be held without guarantee for an agreed upon time. If not guaranteed by that date, the reservation will be dropped.

IEEE-830-1998: Shall be used to aid the creation of this SRS.

1.5 Overview – The key points of this SRS will be the level 0 and level 1 Data Flow Diagrams, a Class Diagram, Use Case Diagram with two common scenarios, and a state diagram.

2. Overall Description

Level 0 Data Flow Diagram:



2.1 Product Perspective – This desktop application is mostly a standalone system that would interface with a privileged B&B employee to input reservation details, communicate to banking servers for transactions, and email servers for notifications.

A) System interfaces –

a) Banking Servers
b) Email servers
c) Printer
d) Card Reader

B) User interfaces –

a) Login screen
b) Calendar with the four rooms on each date.
c) Edit reservation popup menu
d) Employee management menu for accesses.
e) Menu for finance tracking.

C) Hardware interfaces –

a) Windows/MAC PC
b) Database Ethernet/Wireless
c) Printer USBs
d) Card Reader USB

D) Software interfaces –

a) Database software
b) Email software

E) Communications interfaces – Common email and banking protocols.

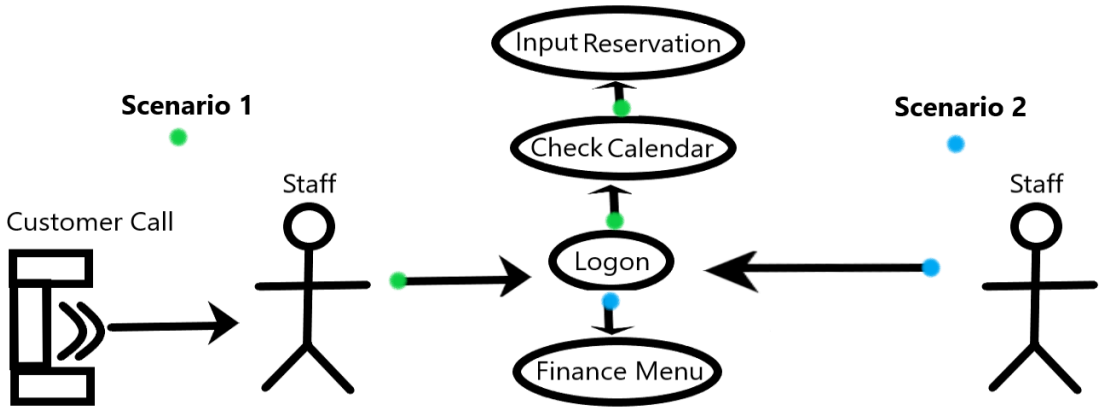
F) Memory – The application itself on the desktop shall be less than a gigabyte in storage. The database of reservations, history, and finances shall start with one terabyte of storage.

G) Operations – A privileged employee will be able to create a reservation, edit the reservation, and input payment information. The application shall be able to process transactions and send reminder emails. The database can be backed up through exports and imports to another hard drive.

H) Site adaptation requirements – The database shall be able to be backed up through exports and imports to another hard drive.

2.2 Product Functions – The desktop application will be able to create a reservation, edit the reservation, and input payment information. The application shall be able to process transactions and send reminder emails. The attached database can be backed up through exports and imports to another hard drive.

Use Case 1.0 Staff Creates a reservation / Wants to check B&B Finances:



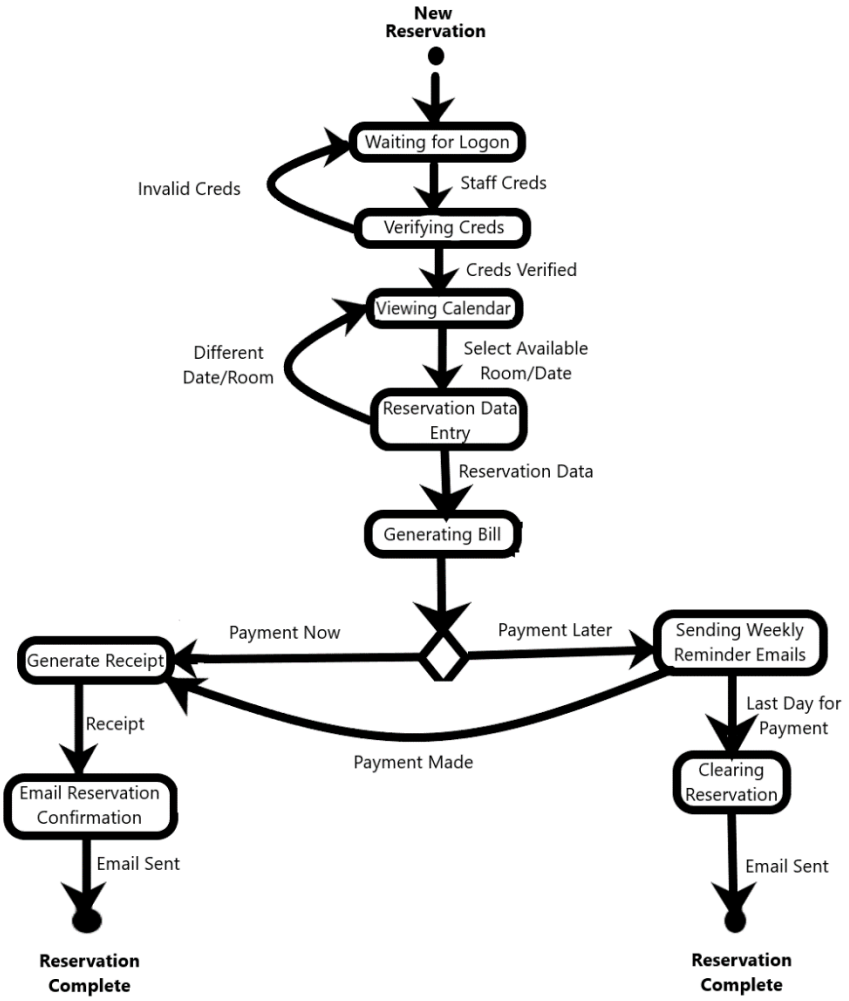
User Scenario 1: Staff Makes Reservation

1. Staff gets a call from a guest
2. Staff logs on to application
3. Staff checks calendar
4. Staff inputs reservation information.

User Scenario 2: Staff Wants to Check Finances

1. Staff Logs on
2. Staff goes to Finance menu

State Transition Diagram:



2.3 User Characteristics – The intended users are solely the B&B staff. Operating the application shall only require some light training of a few hours with minimal technical acumen. Should customers be allowed to make their own reservations, a website can be created based off the desktop application.

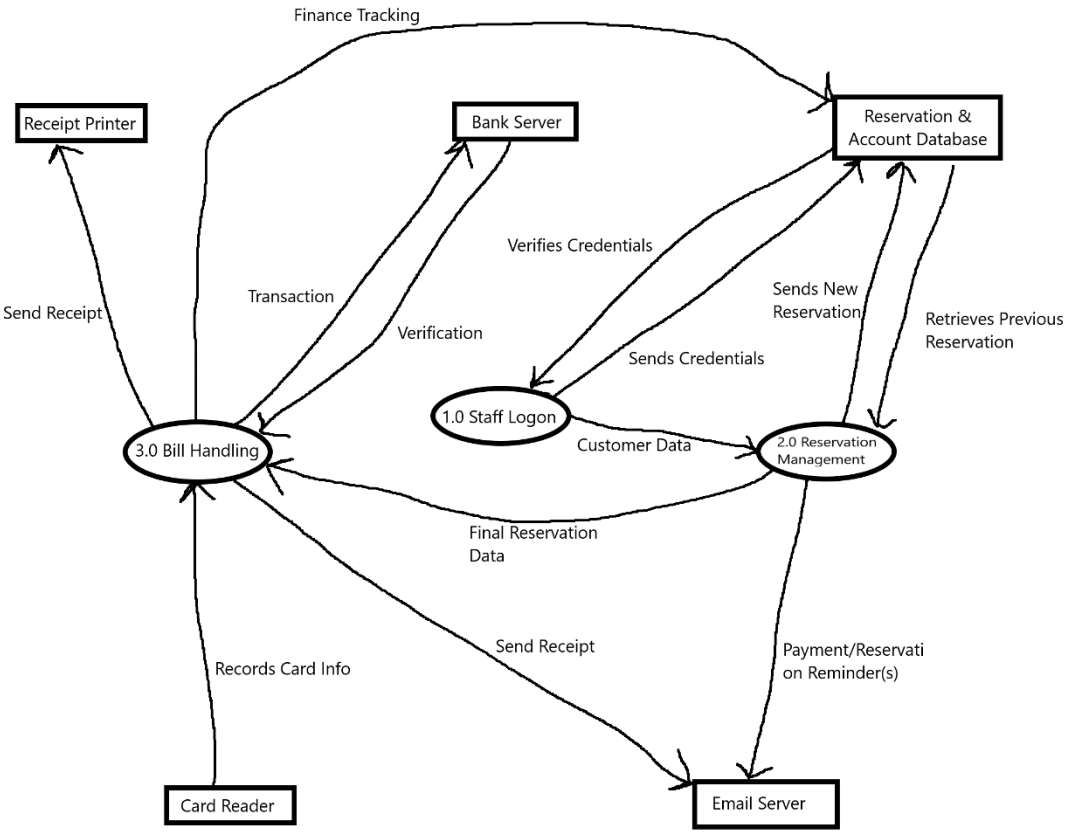
2.4 Constraints – The application shall require an internet connection to process email and banking transactions. The database must be in wired/wireless range of each application installation.

2.5 Assumptions and Dependencies – This desktop application and database setup allows for scalability, but may be reduced to have the database with each installation and synchronized from peer to peer.

2.6 Apportioning of Requirements – Finance metrics to present shall be held for a future build until vetted with customer. Should customers be allowed to make their own reservations, a website can be created based off the desktop application.

3. Specific requirements

3.1 External interface requirements – Level 1 Data Flow Diagram:



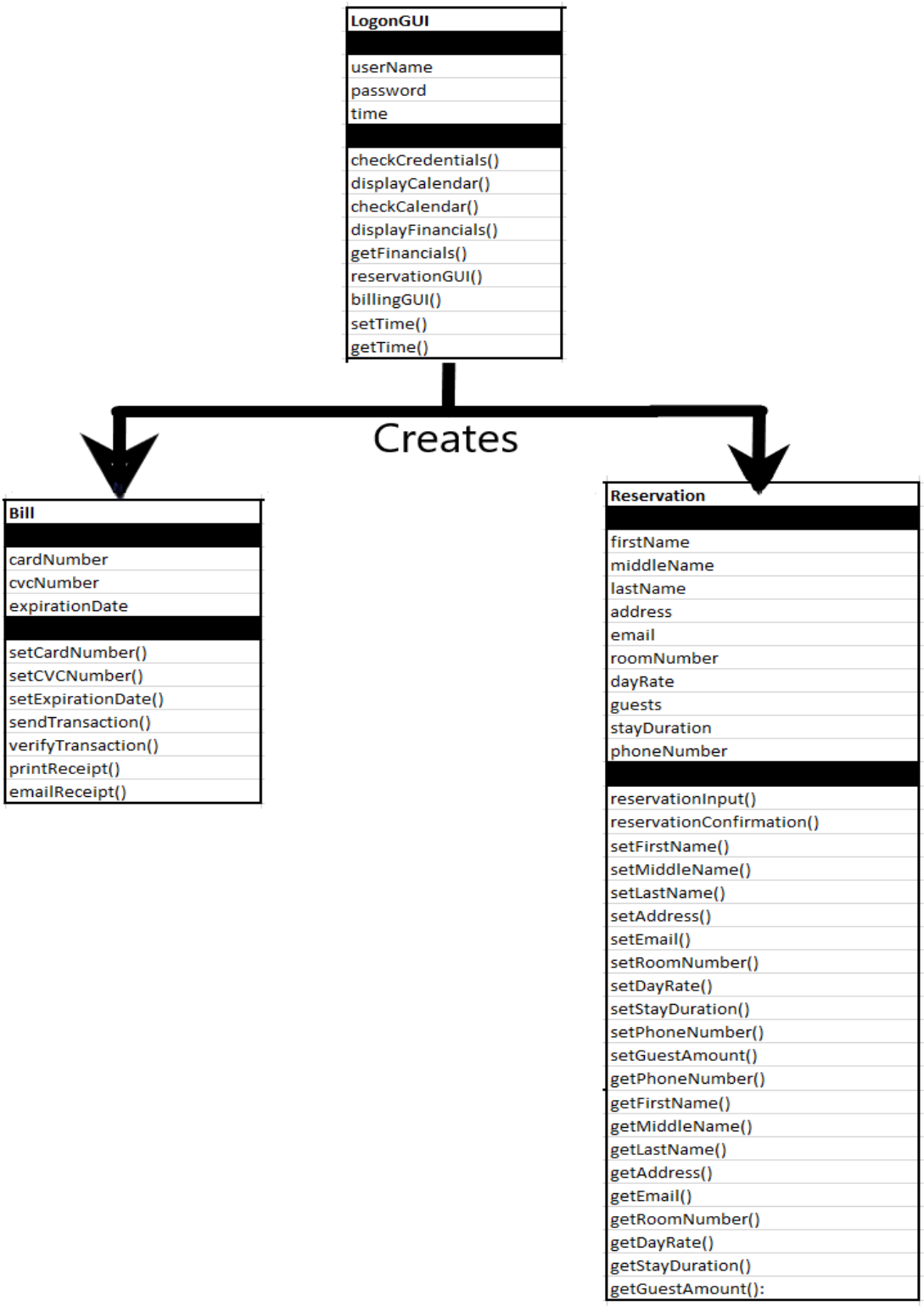
3.1.1 User interfaces – The application shall start with a Login screen. Once user credentials are verified, the application shall move on to a calendar view with current reservations. Each reservation shall have a popup menu when clicked to edit the reservation. Employee management menu for accesses. Menu for finance tracking.

3.1.2 Hardware interfaces - This desktop application shall be able to run on any Windows or Mac PC with modern specifications made past 2010. A wireless or wired connections to the database shall be created depending on B&B schematics.

3.1.3 Software interfaces - The application shall interface with database software such as MySQL/Neo4j. The application shall also interface with preferential email applications such as Gmail and Hotmail.

3.1.4 Communications interfaces - Common email and banking protocols. (SMTP, POP3, IMAP, NTP)

3.2 Classes/Objects – Class Diagram:



3.2.1 Class/Object 1 - LogonGUI

3.2.1.1 Attributes (direct or inherited)

3.2.1.1.1 Attribute 1 – username : Shall be a direct attribute in a name and email format.

3.2.1.1.2 Attribute 2 – password : Shall be a direct attribute stored as a SHA256 hash value, based on the inputted password set by the staff.

3.2.1.1.3 Attribute 3 – time : Direct and inheritable attribute to store the current time.

3.2.1.2 Functions (services, methods, direct or inherited)

3.2.1.2.1 Functional requirement 1.1 – checkCredentials() : Checks username and password with the B&B database.

3.2.1.2.2 Functional requirement 1.2 – displayCalendar(): Displays GUI calendar with clickable reservations and finance menu option.

3.2.1.2.3 Functional requirement 1.3 – checkCalendar(): Checks that inputted reservation does not conflict with other reservations.

3.2.1.2.4 Functional requirement 1.4 – displayFinancials(): Displays financials GUI with metrics based on previous and projected reservations.

3.2.1.2.5 Functional requirement 1.5 – getFinancials(): Imports financial history from the database.

3.2.1.2.6 Functional requirement 1.6 – reservationGUI(): Displays the menu for inputting guest reservation data.

3.2.1.2.7 Functional requirement 1.7 – billingGUI(): Displays the menu for entering guest payment information.

3.2.1.2.8 Functional requirement 1.8 – setTime(): Receives a time from the user's computer.

3.2.1.2.8 Functional requirement 1.9 – getTime(): Sends a time to the Bill or Reservation classes.

3.2.2 Class/Object 2 - Bill

3.2.2.1 Attributes (direct or inherited)

3.2.2.1.1 Attribute 1 – cardNumber : Shall be a direct attribute to hold the full card number of the guest.

3.2.2.1.2 Attribute 2 – cvcNumber: Shall be a direct attribute to hold the 3 digit CVC number.

3.2.2.1.3 Attribute 3 – expirationDate: Shall be a direct attribute to hold the expiration date.

3.2.2.2 Functions (services, methods, direct or inherited)

3.2.2.2.1 Functional requirement 1.1 – setCardNumber(): Records the card number from the billingGUI.

3.2.2.2.2 Functional requirement 1.2 – setCVCNumber(): Records the cvc number from the billingGUI.

3.2.2.2.3 Functional requirement 1.3 - setExpirationDate(): Records the expiration date from the billingGUI.

3.2.2.2.4 Functional requirement 1.4 - sendTransaction(): Sends billing information to a banking server.

3.2.2.2.5 Functional requirement 1.5 – verifyTransaction(): Records transaction confirmation in B&B database.

3.2.2.2.6 Functional requirement 1.6 – printReceipt(): Sends receipt to printer.

3.2.2.2.7 Functional requirement 1.7 – emailReceipt(): Emails receipt to guest(s).

3.2.3 Class/Object 3 - Reservation

3.2.3.1 Attributes (direct or inherited)

3.2.3.1.1 Attribute 1 – firstName : Holds the guest's first name.

3.2.3.1.2 Attribute 2 – middleName : Holds the guest's middle name.

3.2.3.1.1 Attribute 3 – lastName : Holds the guest's last name.

3.2.3.1.2 Attribute 4 – address: Holds the guest's address, this attribute may be omitted.

3.2.3.1.1 Attribute 5 – email: Holds the guest's email address.

3.2.3.1.2 Attribute 6 – roomNumber: Holds the room number to be reserved as indicated from the calendar GUI.

3.2.3.1.1 Attribute 7 – dayRate: Holds the rate a guest would be charged, shall be indicated within the calendar GUI.

3.2.3.1.2 Attribute 8 – guests: Holds the amount of guests to be staying.

3.2.3.1.1 Attribute 9 – stayDuration: Holds the amount of days the guests will be staying.

3.2.3.2 Functions (services, methods, direct or inherited)

3.2.3.2.1 Functional requirement 1.1 – reservationInput() : This function takes in all the attributes mentioned for this class from the log on reservation GUI.

3.2.3.2.2 Functional requirement 1.2 – reservationConfirmation(): This function saves the reservation to the B&B database, and emails the guest a confirmation of their reservation with reminders if the guest chooses to pay at a later date.

3.2.3.2.3 Functional requirement 1.3 – setFirstName(): Handles changing only the first name from the reservation GUI.

3.2.3.2.4 Functional requirement 1.4 – setMiddleName(): Handles changing only the middle name from the reservation GUI.

3.2.3.2.5 Functional requirement 1.5 - setLastName(): Handles changing only the last name from the reservation GUI.

3.2.3.2.6 Functional requirement 1.6 - setAddress(): Handles changing only the address from the reservation GUI.

3.2.3.2.7 Functional requirement 1.7 - setEmail(): Handles changing only the email from the reservation GUI.

3.2.3.2.8 Functional requirement 1.8 - setRoomNumber(): Handles changing only the guest's room number from the reservation GUI.

3.2.3.2.9 Functional requirement 1.9 - setDayRate(): Handles changing only the guest's day rate from the reservation GUI.

3.2.3.2.10 Functional requirement 1.10 - setGuestsAmount(): Handles changing only the amount of guests from the reservation GUI.

3.2.3.2.11 Functional requirement 1.11 - setStayDuration(): Handles changing only the amount of days guests are staying from the reservation GUI.

3.2.3.2.12 Functional requirement 1.12 – setPhoneNumber(): Handles changing only the phone number of the guest from the reservation GUI.

3.2.3.2.13 Functional requirement 1.13 – getFirstName(): Handles changing only the first name for the database.

3.2.3.2.14 Functional requirement 1.14 – getMiddleName(): Handles changing only the middle name for the database.

3.2.3.2.15 Functional requirement 1.15 - getLastName(): Handles changing only the last name for the database.

3.2.3.2.16 Functional requirement 1.16 - getAddress(): Handles changing only the address from for the database.

3.2.3.2.17 Functional requirement 1.17 - getEmail(): Handles changing only the guest's email for the database.

3.2.3.2.18 Functional requirement 1.18 - getRoomNumber(): Handles changing only the guest's room number for the database.

3.2.3.2.19 Functional requirement 1.19 - getDayRate(): Handles changing only the guest's day rate for the database.

3.2.3.2.20 Functional requirement 1.20 - getGuestsAmount(): Handles changing only the amount of guests for the database.

3.2.3.2.21 Functional requirement 1.21 - getStayDuration(): Handles changing only the amount of days guests are staying for the database.

3.2.3.2.22 Functional requirement 1.22 – getPhoneNumber(): Handles changing only the phone number of the guest for the database.

3.2.1.3 Messages (communications received or sent) – Networking communications shall be able to send emails out to guests for reservation confirmation, receipts, and reminders for payment. Banking communications must also be sent and received to confirm a successful transaction.

3.3 Performance requirements – Menu and database operations shall be responsive within three seconds. This is excluding importing and importing a database for backup. Transaction operations with a financial institution shall take no longer than one minute.

3.4 Design constraints – The owner of the B&B may decide on an external database for scalability or peer to peer to save on infrastructure. This desktop application shall be able to run on any Windows or Mac PC with modern specifications made past 2010. A wireless or wired connections to the database shall be created depending on B&B building schematics.

3.5 Software system attributes – Additional drivers may be necessary for the credit/debit card reader and printer. A sufficient GUI library shall be selected with input from the B&B owners.