

John and Jane's Bed & Breakfast Software Design Descriptions

Frontspiece:

Date of issue and status – 12 September 2022, Pending Review

Issuing organization - UMGC

Authorship – Michael Rodriguez

Change history – First Publication

Additional methods to revise Software Requirement Sheet: `accountingGUI()` and `staffPayroll()`, `changeCredentials()`, `getCardNumber()`, `getCVCNumber()`, `getExpirationDate()`.

New Attributes: `staffMember`, salary hours, bonus

Attribute and method class ownership may be referenced in the revised Class Diagram.

Introduction:

Purpose – This document is to provide Data, Architectural, Interface, and Procedural design viewpoints for the John and Jane's Bed & Breakfast Software Application. These viewpoints will be expanded on with a design graphical view and rationale.

Scope- This Software Design Description will cover the inner software implementations and its interactions with outside systems including financial institutions and input/output hardware.

Context – A broader scope of these design descriptions may be found in the corresponding Software Requirements Sheet.

Summary-

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 SDD.

IEEE-1016-2009: Shall be used to aid the creation of this SDD such as with Table 1.

Corresponding Software Requirements Sheet.

Glossary:

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". Username will be shortened to "UN" and password will be referred to as "P."

Identified stakeholders and design concerns – The sole stakeholders for the development of this software application are the B&B owners and their staff. Their design concerns shall be taken into account to make sure the application balances cost, timely delivery, software quality, and functionality.

Data Design Viewpoint

Context - The successful completion of the B&B application shall manage at least twenty attributes and fifty-one methods to function within itself. Most of the data is self-sufficient only requiring outside system inputs from banking servers, card readers, and printers.

Composition - The overarching classes within the application are the LogonScreen, CalendarMenu, ReservationMenu, AccountManagement, and Financial Management that is a part of the LogonGUI system module. GuestReservation and GuestInfo are components of the Reservations system module. Lastly Receipt, BankTransaction, and CardInfo are components of the Billing System Module.

Logical – There are class associations between:

1. LogonScreen and CalendarMenu
2. GuestInfo and GuestReservation
3. Receipt, BankTransaction, and CardInfo Classes.

Dependency – The CalendarMenu is the parent class to the ReservationMenu, AccountManagement, and FinanceMenu classes.

Information – The displayCalendar() method shall be used throughout each child class menu to assist in the generation of the GUI. The getTime() function shall be used to display and calculate times for each menu.

Patterns – A database object can be passed through to each application generated from the B&B database to save on compute time.

Interface – The reservation menu, Financial Menu and Calendar Menu will rely the most on the B&B database.

Interaction / State dynamics – The LogonScreen class manages the time for every other menu. Successful verification is required to continue any menu except the Logon Menu. The Reservation Menu will be using the setters and getters of the GuestInfo and GuestInformation classes. The Receipt class will use the getters of the CardInfo Class. The Reservation Menu will use the setters of the CardInfo class. The Receipt Class shall log the receipt within the emailReceipt() function.

Algorithm – displayFinancials() will calculate all reservation costs once they are imported from the database with getFinancials().

Resources – Verification must be inputted from banking servers, physical card data from card readers, and printer driver information for a likely one-time setup per installation.

Design Rationale:

This Class Hierarchy is designed to make logging organized by Guest, their Reservation(s), and payment information. The menu classes are separated out to focus on their specific GUI elements. It also allows reuse of the time and Calendar GUI elements.

Data Design View:

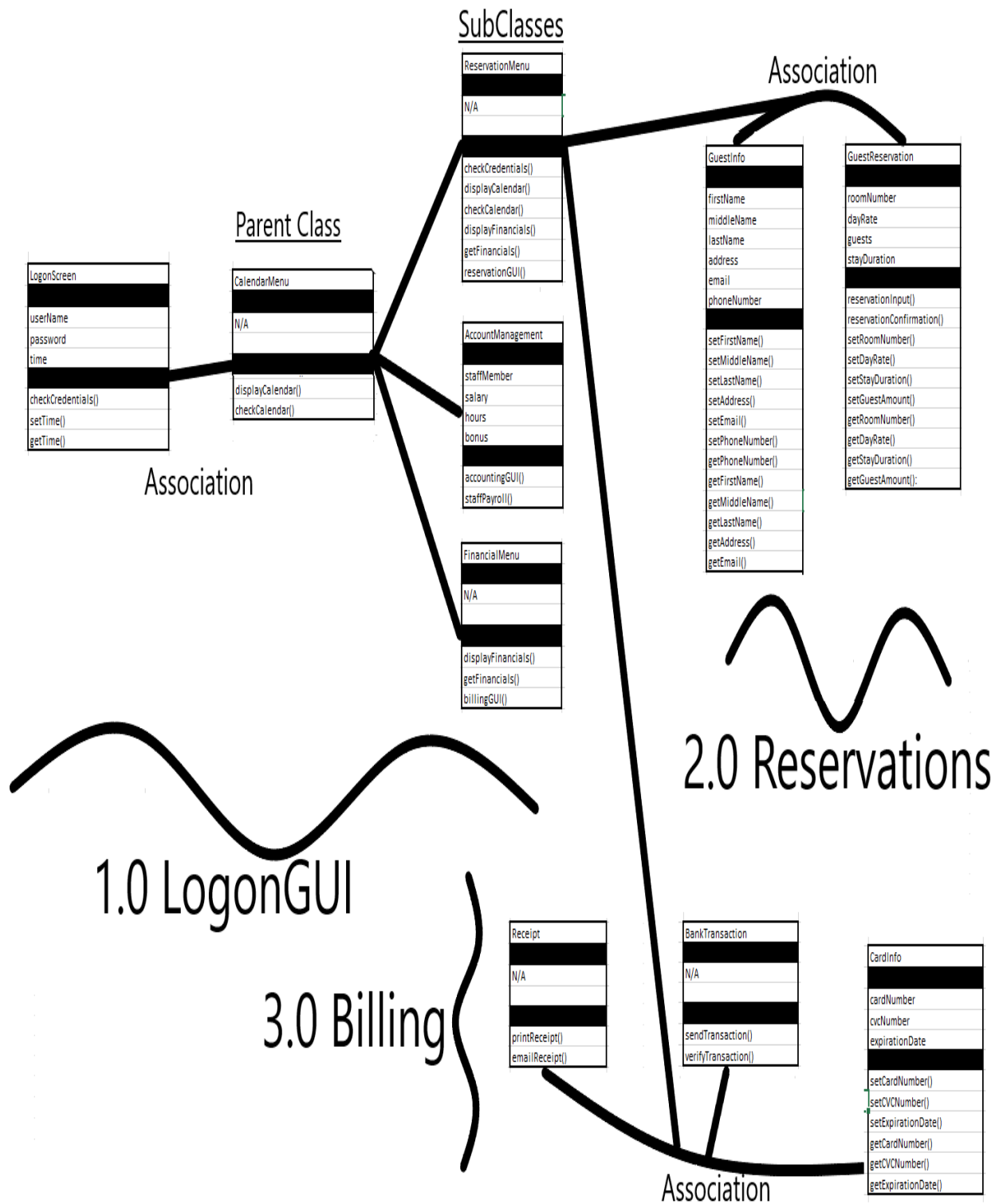


Figure 1: Revised Class Diagram

Architectural Design Viewpoint

Context - The Billing, Reservation, and Logon Modules shall be further broken down into classes that interact with five external systems. The modules can be referenced from the SRS or Data Design Viewpoint.

Composition/Logical - A B&B staff member will typically start from the logon screen, having their credentials be validated by the database. The Calendar Menu and supplemental Financial, Account Management, and Reservation Menus will also need to read and write to the database. This is especially so when handling guest reservation, information objects, and bank transaction objects. Once a reservation is inputted or receipt is generated, the application shall contact an email server to send confirmation emails to the Guest. The Card Information and Receipt objects will have physical interfaces with a receipt printer and card reader if necessary. The Bank Transaction Class shall process the card information object by interacting with the bank server servicing the card.

Dependency - The five external systems for dependency other than the computer itself is the B&B database, an email server, a card reader, printer, and bank server. The receipt printer need not always be used if the guest desires and email only receipt. The usage of digital card information or a card reader is interchangeable.

Information - Guest reservation information is continuously passed from all three main modules, eventually logged in the database.

Patterns - No additional design ideas need be implemented.

Interface - The connection to the database will ideally be hardwired as a priority, but then switched to wireless when not possible.

Interaction - The B&B staff need only concern themselves with the application menus card reader, and printer unless troubleshooting needs to occur between the external systems.

State dynamics - Differing emails shall be sent depending on whether a receipt has been generated yet on successful transaction.

Algorithm - The Financial Menu shall be able to compute total earnings and losses based on Reservation and receipt information from the database. Time based and financial calculations shall also be used in the Account Management Menu from the database.

Resources - The B&B database will need ethernet wiring suitable to the B&B structure. A wireless option may be cheaper and discussed with the B&B owners.

Design Rationale: This architecture design focuses on reservation management and its supplemental external systems. The reservation process with all information read shall take no longer than a minute to process only due to Bank server latency. All other menus shall have load times within three seconds. Completed transactions are then able to instantly be logged and referenced in the Financial Menu.

Architectural Design View:

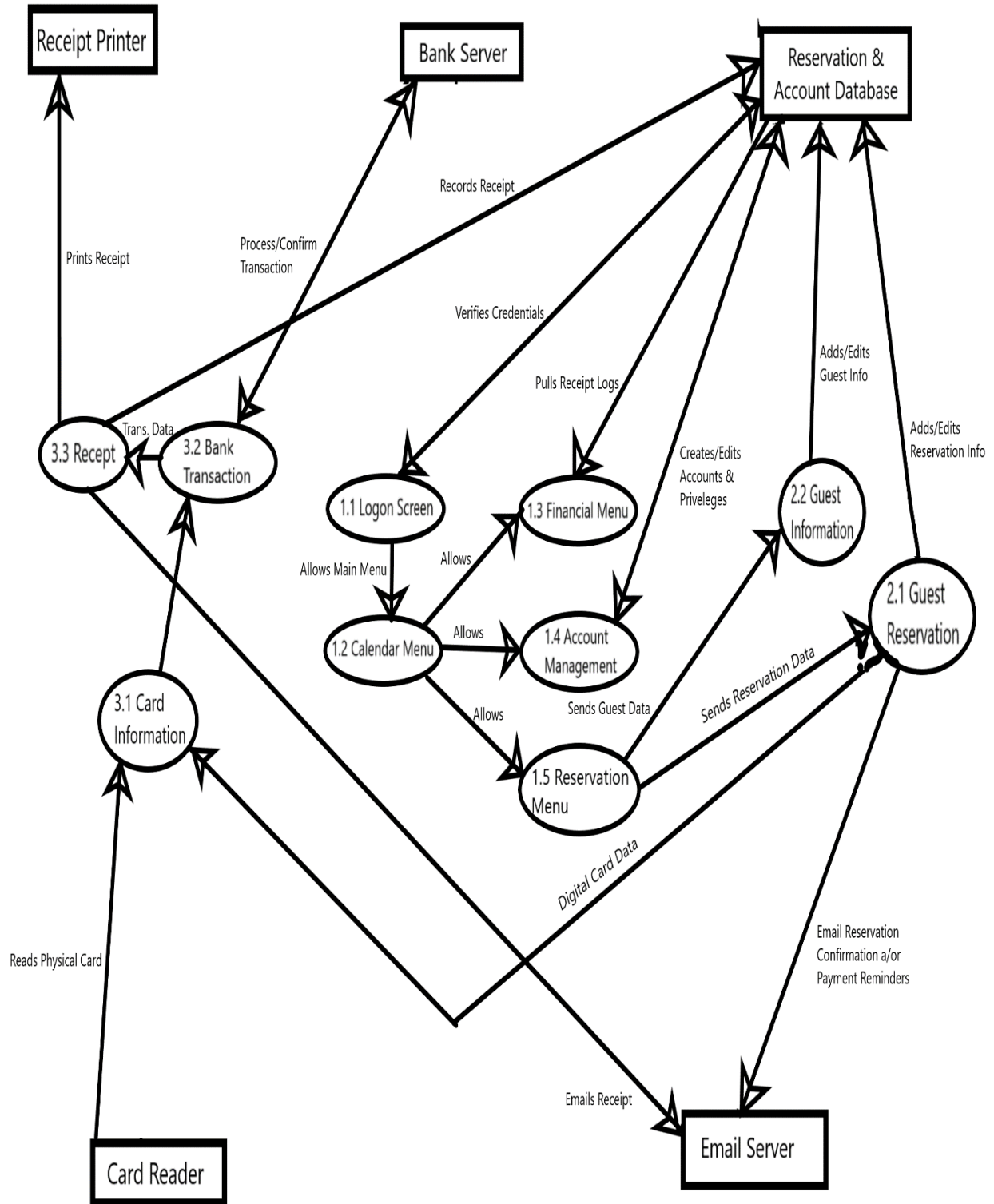


Figure 2: Architectural Control Diagram through level 2 Data Flow Diagram

Interface Design Viewpoint

Context - There are five menu interfaces that can be used by the B&B staff.

Composition - These interfaces are the Logon Screen, Calendar Menu, Reservation Menu, Financial Menu, and Account Menu.

Logical - Each human interactable menu shall be generated from their respective class.

Logon Screen -> 1.1 Logon Screen

Calendar Menu -> 1.2 Calendar Menu

Financial Menu -> 1.3 Financial Menu

Account Menu -> 1.4 Account Management

Reservation Menu -> 1.5 Reservation Menu

Dependency - The Calendar Menu class will provide a calendar object that will be used by the Financial Menu and the Account Menu. The filling in of calendar reservations will be inputted from the B&B database. The calendar object can also be augmented to display financial information and Staff shifts.

Information - The Calendar object is always in display unless covered by a Reservation Menu or Logon Menu.

Patterns - Additional metrics financial and shift metrics can be applied to their respective calendar object to fit the B&B needs.

Interface - Logon Screen usage uses a standard UN and P. The Ps shall be stored as a SHA256 hash in the B&B database. The Calendar Menu after a successful logon will appear showing current reservations in a "by month" format. It will also have a collapsible column for accessing the Account Menu, Financial Menu, and Logout option. The Reservation Menu shall open by clicking on an either empty or filled reservation next to a room to add or edit. The Financial Menu shall provide metrics for the B&B revenue and expense data expressed through the calendar or exported to a spreadsheet. The Account Menu shall provide a Staff List with editable payments and shifts. The calendar in the Account Menu shall also be used to display the work schedule of staff members.

Structure - The Calendar Menu is the root menu from which all other menus except the Logon Menu branch from. The Logon Menu is a precursor to the Calendar Menu for authentication.

Interaction - Staff Employees will be able to access all menus, but not be able to edit their payment information, or overwrite historical records. Staff Leadership will have full functionality to change any information of the B&B database. The Interface shall start scaling up from a 720p monitor resolution.

State dynamics - The Calendar of the Calendar Menu will populate with reservations once the Reservation Menu is completed.

Algorithm - Financial Metrics within the Financial Menu shall be calculated using the day rate, number of guests, and expenses.

Procedural Design Viewpoint

Context - The services provided by this software application shall be focused on Reservation Management and Financial Management. Staff and Account Management are supplemental to software security to ensure intended usage.

Composition - Reservation Management services can be further broken down to creation, editing, and recording guest information. Financial Management services can be broken down to processing customer transactions, Staff payouts, and logging of revenue.

Logical - The reservation procedural logic starts from logon and ends with a reservation confirmation email. The logical steps are as described in Figure 4. Additional logic may be branched off of viewing the calendar to the Financial and Account Menus. A common task may be to edit or check shift information for an employee or export financial data into a report.

Dependency - The successful completion of these processes requires staff to be trained on navigating to the intended interface and recognizing the desired function.

Information - The Calendar Menu will be available for interaction to switch between procedures in three out of the five possible interface menus. Minimization, Maximization, and closing of menus shall allow references to the main calendar and task switching.

Patterns - Emails are sent by the application regardless of payment status. The attributes for first name, last name, email, room#, date range, # of guests, and day rate are required to make a reservation. A default financial metric or shift schedule can be assigned in their respective menus.

Interface - Refer to the Interface Design Viewpoint.

Interaction - Along with interaction with an email server, a banking server will also be contacted once a bill has been generated.

State dynamics - The application will send weekly reminder emails for payment to the potential guest if payment was not made. The reservation is canceled with another email if payment is not received by the day before the scheduled reservation as per the Statement of Need.

Algorithm - No algorithms are needed on a procedural level.

Resources - A printer and card reader are required for physical inputs.

Design Rationale: An authenticated account system is recommended for cyber security, preventing low probability high risk exploitation. Payment reminder emails may be sent out at a differing frequency, but sent weekly to help ensure payment.

Procedural Design View:

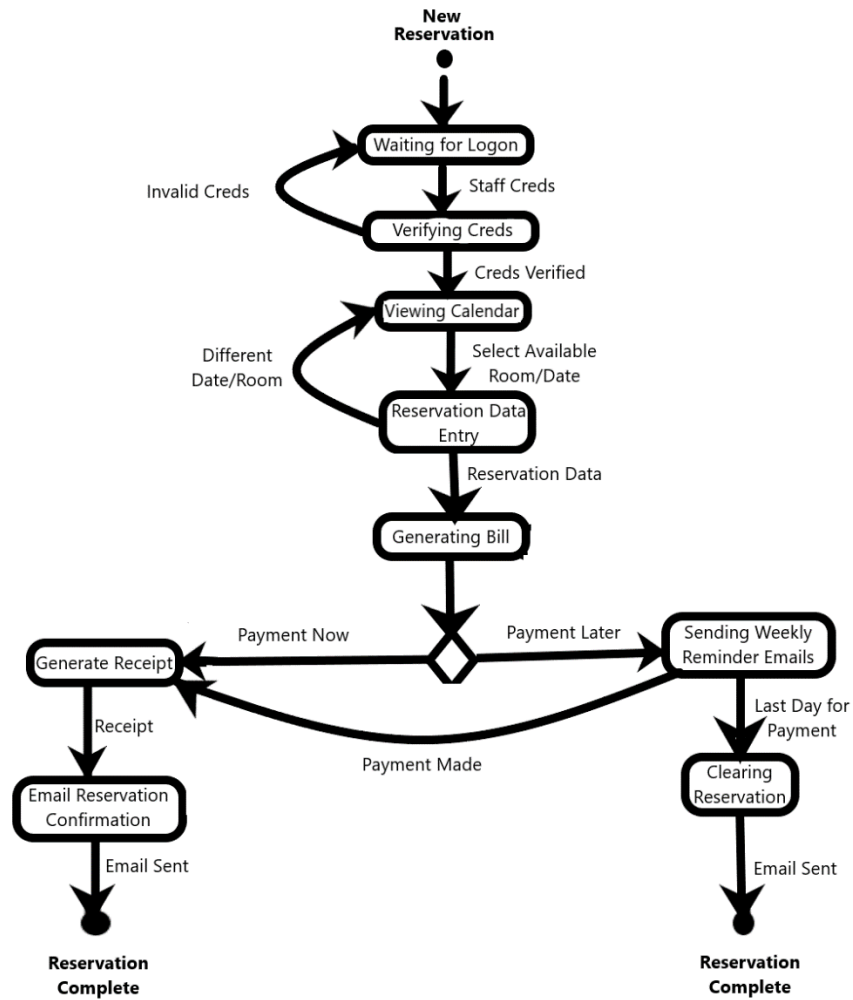


Figure 4: State Transition Diagram for Reservations