**System Design Specification**

Customer and Appointment Tracker

Maddison Kiefer, Steven Rendon, Courtney Sidock

Roland Morales

CS4910A-A1 Senior Project

2/14/2024

**Table of Contents**

Management Summary

**Overview**

The Customer and Appointment Tracker application aims to provide healthcare clinics with a digital solution for managing customer data and appointments efficiently. It includes features such as a daily appointment schedule, client data management, appointment scheduling, and search functionality. The application will be developed using Windows Forms in Visual Studio, with a backend database using SQL Server to store customer and appointment information.

**Target Users**

The primary users of the application are healthcare clinic staff, including doctors, nurses, receptionists, and administrative personnel. The application is suitable for various healthcare settings, including primary care clinics, specialized clinics, hospitals, urgent cares, veterinary clinics, and surgical centers.

**Scope and Timeline**

The scope of the project includes designing and developing the application components, setting up the database, implementing business logic, testing, deployment, and ongoing maintenance. The estimated development time is 165 hours, covering tasks such as database setup (20 hours), GUI design and development (40 hours), business logic implementation (50 hours), and testing procedures (55 hours). The estimated total cost ranges from $10,375 to $10,775, considering development costs, hardware costs, training costs, and contingency.

System Components Details

**Program Design**

The application will be developed using Windows Forms in Visual Studio and will be written in the C# language. SQL queries will be used to access data and input data into the database. For creating the schedule, multiple lists will be used to store the rooms, the appointments, and the colors for highlighting the appointments. The add client and scheduling appointment will include SQL queries to input data to the database. The search functionality will use a data grid to display the information and SQL queries to retrieve the information from the database.

**Output Design**

The output design includes four different pages that users can access to view and complete certain tasks. The first page is the schedule where the user will be able to access all of the other pages and view the scheduled appointments for the day. The schedule will be in the form of a grid with the current date being displayed at the top. There will be previous and next buttons to allow the user to change the date of the schedule. The rooms for the appointments will be labeled at the top of the grid and the times for the appointments, going in hour increments will be displayed along the side. Each appointment that is displayed will be highlighted a different color for easy visibility. At the top of the page there will be buttons to access the ‘Add Client’ and ‘Schedule Appointment’ pages, as well as a search bar to search the database and display the client information.

Clicking on the ‘Add Client button’ will allow the user to access the page to enter in all of the information for a new client. They can enter this information using text boxes and each textbox will have a label above it to clarify the information that should be input. The same goes for the ‘Schedule Appointment’ page. One each of these pages, there will be a button to confirm all of the information and add it to the database.

The search functionality is a search bar where either the first or last name of a client is entered and a separate screen will be displayed, showing all of the clients information in a data grid. This is the main way to access and view the information for the clients.

**Input Design**

The main type of input for the application is from text boxes. Users will enter the data into each text box so that it can be stored in the database. For scheduling an appointment, a DateTimepicker is used so that dates can easily be selected from a calendar, and times can easily be selected using arrows.

Validating the input is done by ensuring that the type of data entered into the text box matches the type of variable that it is being stored as. For example, the age of a client is stored as an integer, so a number should be input. Exception handling will be included to ensure that incorrect input is handled appropriately. If the incorrect input is entered, a message will be displayed letting the user know that the input is incorrect and to enter it in the correct format. Once everything is input correctly, a message will be displayed letting the user know that it has been successfully added to the database.

**File and Database Design**

The database for the application includes tables depending on the clinic that the application is being used for. For a Primary Care Clinic, two tables will be included in the database, one for Customer and one for Appointments. The customer table will include the first name, last name, phone number, email address, date of birth, and address of the customer. The appointment table will include the customer name, the data and time of the appointment, the reason for the appointment, the doctor, the room number, and the length of the appointment. Customers are able to have many appointments so the relationship between the two tables will be a one to many relationship.

The code for the application includes multiple SQL queries that are used to retrieve and store information to and from the database. This allows users to easily access the information in the database from the application.

Regular, daily backups of the database should be done to protect against data loss or potential breaches. In the case of any lost data, the most current backup should be restored to recover as much data as possible.

**Support Processing Design**

The application will not have any background processes of tasks. It will not handle automated messages of notifications. However, detailed exception handling will be included to ensure that all inputs are correct and can be read properly by the system. All of the data is then stored in a database to later be accessed by the application if needed. A logging system will be used to record errors and warnings so that maintenance can be done when needed.

Environmental Requirements

**Hardware:**

Desktop or laptop computers with at least:

Intel Core i5 processor or equivalent for smooth performance.

8GB of RAM to handle database operations and GUI rendering efficiently.

256GB of available storage space for storing application data and files.

Display resolution of 1920x1080 pixels or higher for better readability of the GUI elements.

**Software:**

Windows operating system: Windows 10 Professional (64-bit) recommended for business environments.

.NET Framework: Version 4.8 or later for compatibility with Windows Forms applications.

SQL Server: SQL Server Express 2016 or later for local database hosting, considering the need for three database tables (clients, pets, appointments).

Visual Studio: Visual Studio Professional 2019 or later for development, providing advanced debugging and testing features.

**Infrastructure:**

Network Connectivity: A stable internet connection with at least 2 Mbps bandwidth for accessing cloud-based resources or remote databases. Since the application will primarily be used locally, network stability is crucial for accessing the local database and potential future integration with external services.

Database Hosting: SQL Server Express can be installed locally on the veterinary office's server or a dedicated workstation for centralized data management. Ensure adequate backup procedures are in place to prevent data loss.

Security Measures: Implement role-based access control (RBAC) to restrict access to sensitive data based on user roles (e.g., veterinarian, receptionist). Encrypt communication between the application and the database to protect data privacy.

**Compatibility:**

Test compatibility with popular web browsers to ensure smooth operation if the application is accessed through a web interface in the future.

Ensure compatibility with common veterinary office hardware peripherals such as printers and scanners for document management.

Verify compatibility with veterinary office management software or Electronic Health Record (EHR) systems for potential integration opportunities.

**Scalability:**

Design the application to handle a growing number of clients, pets, and appointments over time. Consider database indexing and optimization techniques for improved performance.

Plan for scalability by designing the application architecture to support future enhancements such as additional features or integration with external systems.

Implementation Requirements

**Development Environment Setup:**

Ensure that each developer has access to the necessary development tools, including Visual Studio and SQL Server Management Studio.

Set up version control systems (e.g., Git) repositories to manage source code and facilitate collaboration among team members.

Establish coding standards and guidelines to ensure consistency and maintainability of the codebase.

**Database Design and Setup:**

Design the database schema with three tables: clients, pets, and appointments, ensuring appropriate normalization and relationships between tables.

Set up the SQL Server database instance, either locally using SQL Server Express or on a dedicated server, and create the necessary database objects (tables, indexes, constraints).

Populate the database with sample data for development and testing purposes.

**GUI Design and Development:**

Design the graphical user interface (GUI) elements, including screens for adding new clients, scheduling appointments, searching for existing clients, and viewing appointments on the calendar.

Develop the GUI using Windows Forms in Visual Studio, ensuring a user-friendly and intuitive layout.

Implement data binding to connect GUI elements to the underlying database tables for seamless data interaction.

**Business Logic Implementation:**

Implement the business logic layer to handle data validation, processing, and manipulation.

Develop functionality for adding new clients, adding pets to clients, scheduling appointments, and searching for existing clients based on user input.

Ensure that the application enforces business rules and constraints to maintain data integrity and consistency.

**Testing Procedures:**

Develop test cases to validate the functionality of each feature and scenario, including positive and negative test cases.

Conduct unit testing to verify the correctness of individual components and modules.

Perform integration testing to ensure that all components work together seamlessly.

Conduct user acceptance testing (UAT) with stakeholders to validate that the application meets their requirements and expectations.

**Deployment Plan:**

Prepare deployment packages for distributing the application to end users, including setup/installation files and documentation.

Define the deployment process, including steps for installing the application on users' machines and configuring database connections.

Provide user training and support materials to help users get started with the application effectively.

**Maintenance Strategy:**

Establish procedures for ongoing maintenance and support, including bug fixes, updates, and enhancements.

Set up a system for logging and tracking issues reported by users and implementing timely resolutions.

Plan for future updates and iterations based on user feedback and evolving requirements.

Time and Cost Estimates

**Time Estimate**

Development Time:

Database Design and Setup: 20 hours

GUI Design and Development: 40 hours

Business Logic Implementation: 50 hours

Testing Procedures:

Unit Testing: 20 hours

Integration Testing: 15 hours

User Acceptance Testing: 20 hours

Deployment Plan: 10 hours

Training Time: 10 hours

Total Development Time: 165 hours

**Cost Estimate**

Development Costs: Assuming an average hourly rate of $50 for developers:

Development Time (165 hours): $8,250

Software Costs:

Visual Studio Community (free): $0

SQL Server Express (free): $0

Hardware Costs: A standard desktop computer with sufficient processing power, memory, and storage capacity to run the application and host the database.

Cost: $800 - $1,200 (depending on specifications)

Training Costs: Assuming an average hourly rate of $50 for trainers:

Training Time (10 hours): $500

Miscellaneous Costs: (10% of total development cost as contingency)

Contingency: $825

Total Estimated Cost:

Development Costs: $8,250

Hardware Costs: $800-$1200

Training Costs: $500

Contingency: $825

*Estimated Total: $10,375-$10,775*

These estimates provide a basic framework for the time and cost required to develop and deploy the veterinarian office customer and appointment tracker application. It's important to note that these are rough estimates and may vary depending on factors such as team experience, project complexity, and unforeseen challenges. Additionally, actual costs may differ based on specific rates and expenses in your context.

Conclusion

The System Design Specification lays out the plan and overall design for the Appointment Tracker and Customer Database Application. It covers all of the component details such as the input design, the database design, and the program design so that readers can get a better idea of how the application is going to work. It also covers the time and cost estimates of this type of project and covers the environmental requirements and implementation requirements. This documentation helps to give readers a much better understanding of what is planned for this application and how it will function once it is developed.