Tesseract Software Designs

Arch Dental Associates

Adam Franzen

COSC 6000 FA2 2016

Contents

[The Problem 4](#_Toc465894127)

[Background Information 4](#_Toc465894128)

[Problem Description 4](#_Toc465894129)

[Users / Roles 4](#_Toc465894130)

[Requirements 5](#_Toc465894131)

[Functional 5](#_Toc465894132)

[Non-Functional 6](#_Toc465894133)

[User Information 7](#_Toc465894134)

[Site Diagram 7](#_Toc465894135)

[Features and Benefits 8](#_Toc465894136)

[UI Screen Prints 8](#_Toc465894137)

[Screen Prints / Reports 8](#_Toc465894138)

[Project Design 9](#_Toc465894139)

[ER Diagram 9](#_Toc465894140)

[Database Schema 9](#_Toc465894141)

[Data Dictionary 10](#_Toc465894142)

[Use Case Diagram (UML) 10](#_Toc465894143)

[Use Cases (Essential Form) 10](#_Toc465894144)

[Use Cases (Expanded Form) 10](#_Toc465894145)

[Domain / Concept Model (UML) 10](#_Toc465894146)

[System Sequence Diagrams (UML) 10](#_Toc465894147)

[Collaboration Diagrams (UML) 10](#_Toc465894148)

[State Chart Diagrams (UML) 10](#_Toc465894149)

[Deployment Diagram (UML) 10](#_Toc465894150)

[Glossary 10](#_Toc465894151)

[Program Implementation 11](#_Toc465894152)

[Program Structure 11](#_Toc465894153)

[Program Code / Comments 11](#_Toc465894154)

[Project Management Section 11](#_Toc465894155)

[Weekly Status Reports 11](#_Toc465894156)

[Activities And Time 11](#_Toc465894157)

[Total Time Spent 11](#_Toc465894158)

[Project Management 11](#_Toc465894159)

[Difficulties Encountered 11](#_Toc465894160)

[Technical Reports Used To Solve Difficulties 11](#_Toc465894161)

[Other Technical Information 11](#_Toc465894162)

[Electronic Documentation 12](#_Toc465894163)

[Computer Disks And All Materials On CD 12](#_Toc465894164)

[Project Materials 12](#_Toc465894165)

[Presentation Materials (Powerpoints, etc) 12](#_Toc465894166)

[Weekly Status 12](#_Toc465894167)

[End of Week 1 12](#_Toc465894168)

[End of Week 2 12](#_Toc465894169)

# The Problem

## Background Information

Arch Dental Associates (from now on referred to as “Arch”) has approached us to design a modern web interface that can help them to coordinate their Dental Practice and make their business more efficient. They are looking to replace an age-old paper system of operating the business. It will be of upmost importance that the user interface for this application be highly user friendly and intuitive to use given that many of the users will have limited computer experience.

## Problem Description

Arch would like to move to a modern/computerized system to allow their patients access to some portion of their account from home, to ensure rooms in each clinic are adequately utilized, and not double-booked, to ensure the time of each provider is adequately utilized, and not double-booked, and lastly, to provide a paperless (or at least significantly less paper) method for day-to-day business and accounting.

## Users / Roles

* Client (Patients)
  + Primary account holder
  + View/edit household account info
  + View/cancel appointments
  + Payments
  + Services
  + Designate primary insurance holder
  + Add/edit dependents
* Office Staff
  + An office worker or provider/dentist
  + Create/view/update individual client account info
  + Create/view/update appointments
  + Process payments
* System Administrator
  + Main owners of practice
  + Create/view/update any screen
  + Create/view/update office staff and providers
  + Create/view/update clinics
  + Create/view/update rooms
  + Create/view/update services
  + Create/view/update qualifications
  + Create/view/update accepted insurance

# Requirements

## Functional

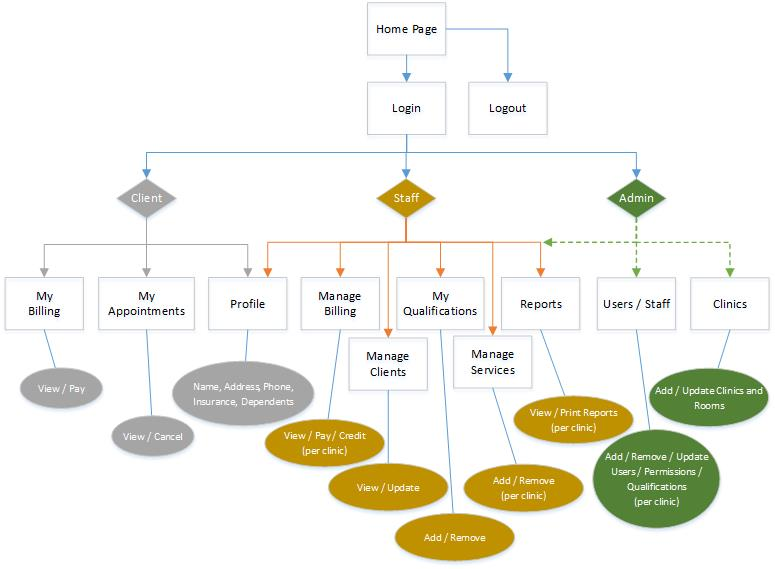
* The system shall support the following users: Client, Office, Admin
  + Client
    - Patients
    - The system shall designate a Primary account holder (a head of household)
    - The system shall allow the Client to view/edit household info such as:
      * Contact Information (Address, Phone)
      * Insurance Information
      * Dependent Information and Relationship to Primary
      * Appointments coming / past
        + Payments due / made
        + Services rendered
      * Designate who is the Insurance holder
      * Designate the Household Insurance Company
      * Allow a Household to be served without insurance
    - The system shall provide a user with a unique account number
  + Office
    - Office Staff, Doctors, Hygienists
    - The system shall allow all staff to create / update any Client account info
    - The system shall allow all staff to create / update appointments
    - The system shall allow all staff to process payments / refunds
    - The system shall allow all staff to alter standard pricing for an appointment (before and after)
  + Admin
    - Main owners/partners of practice (could also be Doctors)
    - The system shall allow Admin to create/view/update any screen
    - The system shall allow Admin to create/view/update office staff / providers
    - The system shall allow Admin to create/view/update clinics
    - The system shall allow Admin to create/view/update rooms
    - The system shall allow Admin to create/view/update services
    - The system shall allow Admin to create/view/update qualifications
    - The system shall allow Admin to create/view/update qualifications assigned to provider
  + All Users
    - The system shall capture identifying information for all users of the system including: Name, Contact Information, Birthdate
    - The system shall provide a default username / password and allow user the ability to change either one
    - The system shall provide a unique numeric ID to each user
    - The system shall encrypt all passwords
* The system shall keep track of all services (and costs for each service) a clinic provides.
  + A provider does not need to be qualified in order to perform a service the clinic offers
  + Designate a standard rate for each service
  + Designate a standard length of time to complete each service
* The system shall allow for more than one dentist and/or hygienist to provide services
* The system shall record services each provider is certified for
* The system shall allow patients to make appointments for services on a specific time/date
* The system shall allow a patient to receive services from multiple providers during a visit
* The system shall track if an appointment is kept/missed/canceled
* The system shall, for each appointment, reserve a room for each service provided, note individual service lengths
* The system shall generate a single bill for an entire household
* The system shall allow Office user creating an appointment to alter standard pricing with billing reflecting both the standard and actual rates
  + The actual rate should never be higher than the standard rate
* The system shall provide the ability to generate an overall bill for all services outstanding, or provide an itemized bill for an individual appointment
  + Should contain the date of service, list of services, name of provider, and costs
* The system shall generate a bill on a monthly basis for each household
* The system shall generate a report containing all households, names, addresses, phones, person ID, dependents and relationships
* The system shall generate a report showing insurance coverage for all households including household ID, household primary account holder name, insurance company ID and name.
* The system shall generate a report containing all patients in order by person ID, name, birthdate, and also contain their insurance company and policy number
* The system shall generate a report show itemized billings for all households with household ID, primary account holder name, person ID, patient name, service, and cost ordered by primary account holder name, patient name, and billing date
* The system shall generate a report showing the total cost of all services received for each household
* The system shall generate a report showing each provider and services qualified to render
* The system shall generate a report all future appointments sorted by patient name, appointment date/time, length of service, phone – grouped by each patient
* The system shall generate a report list all services performed by each provider in order by provider name and include the service ID, service description, and cost
* The system shall generate a report, that for a given date, lists the total amount of services each provider rendered – ordered by provider name

## Non-Functional

* The system shall work only via web browser
* The system shall be optimized for the Google Chrome web browser
* The system shall have a Web API back end written in C# to support .NET 4.0
* The system shall have an HTML5 front end written in Angular JS
* The system shall have a modern/organized/consistent look and feel
* The system shall take into account multiple users working at the same time in order to best optimize for performance
* The system shall leverage an Oracle database for data storage
* The system shall support Microsoft IIS
* The system shall log payment information, but will not actually interface with payment processing companies

# User Information

## Site Diagram



## Features and Benefits

|  |  |
| --- | --- |
| ***Feature*** | ***Benefit*** |
| Calendar accessible from any computer | * No more confusing, disorganized, and not up-to-date paper calendar to keep track of * Clients can view/cancel appointments online |
| Payments tracked by computer | * No more digging through piles of receipts/papers to balance the books * Clients can make payments online |
| Real-world, up-to-date data | * No more over/under-booked rooms * No more over/under-booked providers |
| Clients can manage their data online | * Lighter workload on employees when clients use online services   + Lower call instances for payments and billing questions   + Lower call instances for appointment time/date/service/cancellation questions |
|  |  |

## UI Screen Prints

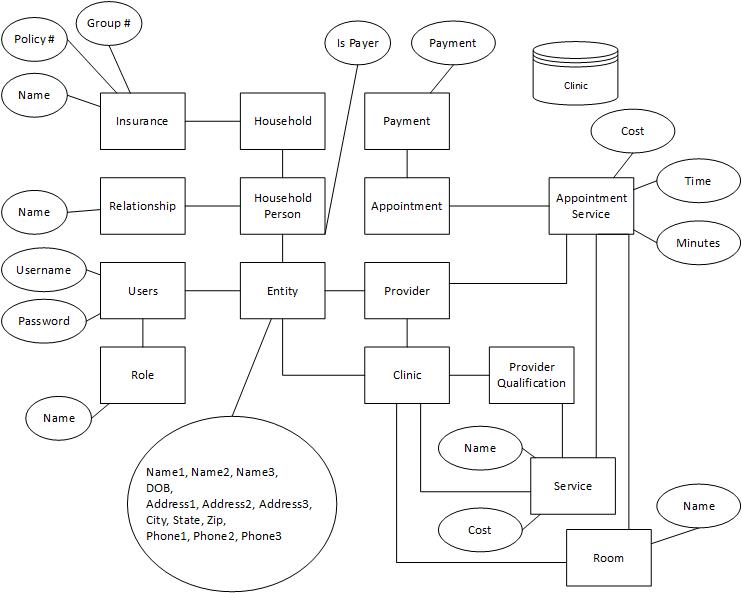
TODO

## Screen Prints / Reports

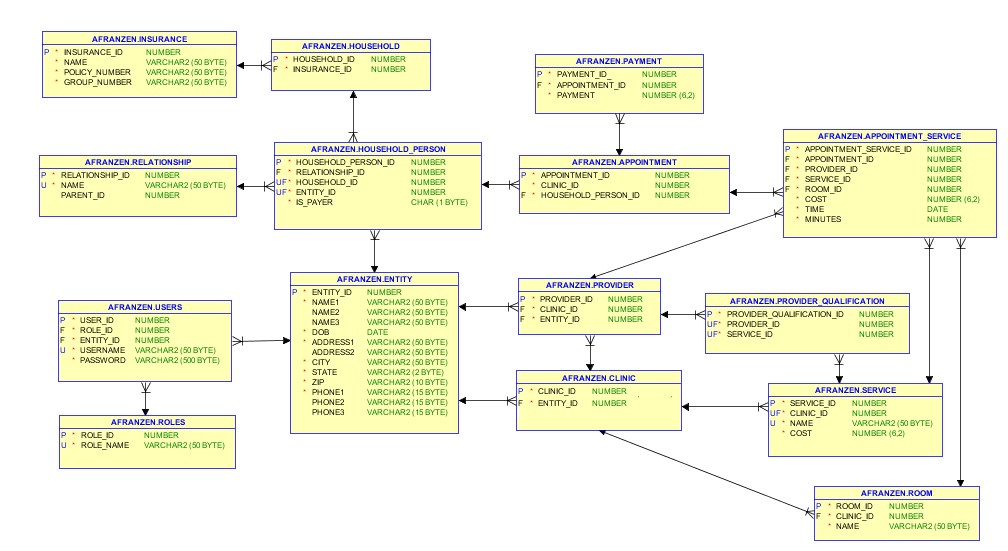
TODO

# Project Design

## ER Diagram



## Database Schema



## Data Dictionary

TODO

## Use Case Diagram (UML)

TODO

## Use Cases (Essential Form)

TODO

## Use Cases (Expanded Form)

TODO

## Domain / Concept Model (UML)

TODO

## System Sequence Diagrams (UML)

TODO

## Collaboration Diagrams (UML)

TODO

## State Chart Diagrams (UML)

TODO

## Deployment Diagram (UML)

TODO

## Glossary

TODO

# Program Implementation

## Program Structure

TODO

## Program Code / Comments

TODO

# Project Management Section

## Weekly Status Reports

TODO

## Activities And Time

TODO

## Total Time Spent

TODO

# Project Management

## Difficulties Encountered

TODO

## Technical Reports Used To Solve Difficulties

TODO

## Other Technical Information

TODO

# Electronic Documentation

## Computer Disks And All Materials On CD

TODO

## Project Materials

TODO

## Presentation Materials (Powerpoints, etc)

TODO

# Weekly Status

## End of Week 1

* Last Week
  + Studied requirements (1 hour)
  + ER Diagram 1st Draft (1 hour)
  + Considering possible UI design (2 hours)
  + Considering back end design (0.5 hours)
  + Research Entity Framework (3 hour)
* This Week
  + Finalize database design
  + Finalize front-end and back-end design
  + Research Entity Framework
  + Write requirements
  + Finalize Users/Roles
* Challenges
  + Had some slip-ups when analyzing the requirements

## End of Week 2

* Last Week
  + Finalized database design (3 hours)
  + Research Entity Framework (4 hours)
    - Throwing out Entity Framework – Non-functional for .NET 4.0
  + Wrote requirements based upon initial requirements doc (1.5 hours)
  + Finalized Users/Roles (0.5 hours)
  + Finalized front-end framework (0.5 hours)
    - Single Page Application, Angular JS, Bootstrap-UI, Angular-UI, LESS, AJAX-JSON
  + Finalize back-end framework (0.5 hours)
    - Microsoft Web API
    - Layered approach (Controller / Business Layer / Data Layer)
    - Data Layer will utilize inline SQL
  + Linked app code to Github for source control and project management
* This Week
  + Begin implementing features
    - Establish UI Framework and tooling
    - Establish back end framework layers and achieve Oracle connection
    - Landing Page
    - Logo
    - Login / Security
    - Navigation
    - Encryption / Decryption of passwords and Tax ID (Social Security Number)
* Challenges
  + Entity Framework
    - Struggled a lot trying to get it to work despite many tutorials working successfully
    - Research shows that Oracle and Entity Framework don’t play nicely in .NET 4.0