

Final Report: ***Chocoholics Anonymous***

CIS 375

Last Updated: 8/20/18

Version 1.0

Team "2"007 Britney Spears:

***Joshua Attard, Veeram Hirekhan, Hassan Mehdi,
Maram Mohammed, Allison Ramasami, Srinivas Simhan***

Table of Contents

Summary of Software Requirements Analysis Specification	3
Summary of Design Document	3
Summary of Test Results	4
Code from Rapid Prototype (SQL Database)	5

Summary of Software Requirements Analysis Specification

The software requirements analysis specification document fully described the expected behavior of the ChocAn software system. Actors and their functions were identified. Robustness diagrams to model the actors and data were created for each use case specification, along with data flow diagrams and sequence/collaboration diagrams. Both functional and non-functional requirements were modeled, and attached were a data dictionary and traceability matrix.

Summary of Design Document

In the Design Document, an architecture diagram is presented for the system design, the details of all the functions are provided for the detailed design, and a decomposition diagram is presented to connect the two. For each function in the detailed design, the class the function is a part of, its interface, its functionality, and the function's side effects are given. In addition, class diagrams, the build ERD, data flow diagrams, a database schema, and all the software layouts are provided to fully flesh out the design of the software.

Summary of Test Results

Test Number	Test Type	Input	Expected Output and/or Action	Actual Output and/or Action	Pass/Fail Decision
1	Black Box Testing	Make a report and fill out everything inside to have a valid report to send.	Should receive a report Friday evening and should be able to read content inside of it.	Reports get generated and contents are readable, but no time trigger.	Fail
2	White Box Testing	Generate the report.	Reports are generated.	Reports are generated.	Pass
3	Black Box Testing	Select items that the member received.	System should output the balance due based off services given.	System outputs balance.	Pass
4	Black Box Testing	Provider puts in invalid member number	System prompts the provider to put in a new member number.	Page layout completed, but doesn't work.	Fail
5	Black Box Testing	Insert username and password. Or swipe/scan card	Enter the system to put information.	Page layout completed, but doesn't work.	Fail
6	White Box Testing	No input, just generate provider report.	Have special report stating that they did not come in a certain week.	No generated report implemented at this time.	Fail
7	Integration Testing	Member signing in with member ID	Ensure that members are correctly suspended or not suspended	Page layout completed, functionality needs work.	Fail

Code from Rapid Prototype (SQL Database)

use chocoholicsanonymous

go

```
CREATE TABLE [Member] (  
    [MemberNumber] NUMERIC(9,0) NOT NULL ,  
    [Name] VARCHAR(25) NOT NULL ,  
    [Street] VARCHAR(25) NOT NULL ,  
    [City] VARCHAR(14) NOT NULL ,  
    [State] CHAR(2) NOT NULL ,  
    [ZipCode] NUMERIC(5) NOT NULL ,  
    [Email] VARCHAR(50) NOT NULL ,  
    [Status] NUMERIC(1,0) NOT NULL , -- what data type should this be?  
    CONSTRAINT [PK_Member]  
        PRIMARY KEY CLUSTERED ([MemberNumber] ASC)  
)
```

```
CREATE TABLE [Provider] (  
    [ProviderNumber] NUMERIC(9, 0) NOT NULL ,  
    [Name] VARCHAR(25) NOT NULL ,  
    [Street] VARCHAR(25) NOT NULL ,  
    [City] VARCHAR(14) NOT NULL ,  
    [State] CHAR(2) NOT NULL ,  
    [ZipCode] NUMERIC(9, 0) NOT NULL ,  
    [Email] VARCHAR(50) NOT NULL ,  
    CONSTRAINT [PK_Provider]  
        PRIMARY KEY CLUSTERED ([ProviderNumber] ASC)  
)
```

```
CREATE TABLE [Service] (  
    [ServiceCode] NUMERIC(6, 0) NOT NULL ,  
    [Name] VARCHAR(20) NOT NULL ,  
    [Fee] DECIMAL(5,2) NOT NULL ,  
    CONSTRAINT [PK_Service]  
        PRIMARY KEY CLUSTERED ([ServiceCode] ASC)  
)
```

```
CREATE TABLE [ProvidedService] (  
    [MemberNumber] NUMERIC(9,0) NOT NULL ,  
    [ProviderNumber] NUMERIC(9,0) NOT NULL ,  
    [ServiceCode] NUMERIC(6,0) NOT NULL ,  
    [DateProvided] DATE NOT NULL ,  
    [DateReceived] TIMESTAMP NOT NULL ,  
    [Comment] VARCHAR(100),  
    CONSTRAINT [FK_ProvidedService_MemberNumber]  
        FOREIGN KEY([MemberNumber]) REFERENCES [Member] ([MemberNumber]),  
    CONSTRAINT [FK_ProvidedService_ProviderNumber]  
        FOREIGN KEY([ProviderNumber]) REFERENCES [Provider] ([ProviderNumber]),  
    CONSTRAINT [FK_ProvidedService_ServiceCode]  
        FOREIGN KEY([ServiceCode]) REFERENCES [Service] ([ServiceCode])  
)
```

```
CREATE TABLE [Transaction] (  
    [TransactionNumber] NUMERIC(9, 0) NOT NULL ,  
    [MemberNumber] NUMERIC(9, 0) NOT NULL ,  
    [Amount] DECIMAL(5,2) NOT NULL ,  
    [Date] TIMESTAMP NOT NULL ,  
    CONSTRAINT [PK_Transaction]  
        PRIMARY KEY CLUSTERED ([TransactionNumber] ASC),  
    CONSTRAINT [FK_Transaction_MemberNumber]  
        FOREIGN KEY([MemberNumber]) REFERENCES [Member] ([MemberNumber])  
)
```

```
CREATE TABLE [Employee] (  
    [EmployeeNumber] NUMERIC(9, 0) NOT NULL ,  
    [Name] VARCHAR(25) NOT NULL ,  
    [PhoneNumber] NUMERIC(10, 0) NOT NULL ,  
    [Street] VARCHAR(25) NOT NULL ,  
    [City] VARCHAR(14) NOT NULL ,  
    [State] CHAR(2) NOT NULL ,  
    [ZipCode] NUMERIC(5, 0) NOT NULL ,  
    [Email] VARCHAR(50) NOT NULL ,  
    CONSTRAINT [PK_Employee]  
        PRIMARY KEY CLUSTERED ([EmployeeNumber] ASC)  
)
```