

CSC 190

Spring 2013

Team Sierra

Alex Chernyak

Joubin Jabbari

Kyle Matz

Mike McParland

Scott Livingston

Serge Lysak

Software Requirement Specifications

Sunday May 5th, 2013

1. INTRODUCTION

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

1.3.2 Acronyms

1.3.3 Abbreviations

1.4 References

1.5 Overview of Contents of Document

1.5.1 General Description

1.5.2 Specific Requirements

1.5.3 Appendix A - Data Dictionary

2. General Description

2.1 Product Perspective

2.2 Use Case Models of System Features

2.3 Characteristics

Users

Features: Client can

Features: Stylist can...

Features: Admin can...

2.4 General Constraints

2.5 Assumptions and Dependencies

3. Specific Requirements

3.1 Use Case Specifications

3.2 Performance Requirements

3.3 Design Constraints

3.4 Quality Attributes

3.4.1 Reliability

3.4.2 Maintainability

3.4.3 Program Quality Attributes

3.4.4 Security

3.4.5 Transferability / Conversion

3.4.6 Operational Quality Attribute

3.4.7 Operations

3.4.8 Site Adaptations

4. Approvals

Project Sponsors

Advisor

Team Sierra

Appendix A - Data Dictionary

Data Element:

Data Tables:

1. INTRODUCTION

This is the software system proposal document for the “Salon Scheduling System” project sponsored by Dragonfly Salon and Boutique. This project is being undertaken by the “Team Sierra” development team. The team is comprised of undergraduate students majoring in Computer Science at California State University, Sacramento. The team members are enrolled in a two-semester senior project course required of all undergraduate majors. Successful delivery of the desired software product will fulfill the senior project requirement for the student team members.

PROJECT SPONSOR

Contact Person’s Name: Alayna and Lisa Sigurdson

Title: Owners and Hair Stylists

Organization Name: Dragonfly Salon and Boutique

Contact information: Alayna.Sigurdson@gmail.com

DEVELOPMENT TEAM:

“Team Sierra”:

- Alex Chernyak
- Joubin Jabbari
- Kyle Matz
- Mike McParland
- Scott Livingston
- Serge Lysak

Team Contact Info: TeamSierra@googlegroups.com

1.1 Purpose

The propose of this document is to clearly express and reiterate the sponsor needs. In this document, Team Sierra will provide a detail understanding of the requirements. By reading this document, the sponsor should have a clear understanding of what will the finish product will entail. This is a change for the sponsor to correct or confirm miscommunications regarding requirements.

1.2 Scope

In this document, Team Sierra will outline its understanding of what the sponsor's needs are. Based on these needs, use cases will be developed to cover the scope of the operation of the business. Diagrams will be developed to simulate such interactions. Resulting in a clear understanding on the behalf of the sponsor of how this system will be used.

1.3 Definitions, Acronyms and Abbreviations

This section will include the various definitions, acronyms, and abbreviations that we will use throughout this document. This section can be referenced throughout reading the document to further explain unclear words.

1.3.1 Definitions

Actor	Anything that interfaces with the system
App Server	A server that runs the web application.
Database	Organized collection of data.
Entity	An organized object describing related data in the database.
Executable Code	The form of the software that can be ran on the computer.
Faculty Advisor	The senior project teams have a CSc faculty member assigned to them to advise the team on various aspects throughout the course of the project.
Gigabyte	A large unit of data storage.
Javadoc	Javadoc is a documentation standard that generates HTML files with descriptions of the code to be view using any web browser.
MySQL	A database program used to hold and organize data in a relational form.
Project Charter	Provides a detailed approach to the project.
Project Documentation	Includes all of the following documents that will be created for the software system over the course of the project.
Project Management Plan	Provides detailed information on how the project will be managed and resources will be obtained.
Sanitation	Process of validating and removing invalid input data.
Software Design Specification	Provides a high level view of the design and interfaces that the project is based on.
Software Requirements Specifications	Describe the behavior, scope and dependencies of the system to be developed. Provide explanation on how it will be used.
Source Code	A collection of computer instructions written in human-readable computer language.
Stylist	Refers to both hair stylist and nail technician

System Test Specification	Sequence of tests to be conducted to test the system.
System Test Report	Results from the System Test Specification.
Use Case	Describes things that an actor wants to do to the system.
User Manual	A complete guide on how to install and operate the system.

1.3.2 Acronyms

CD	Compact Disk
CSc	Computer Science
ERD	Entity Relation Diagram
GB	Gigabyte
IDE	Integrated Development Environment
SPMP	Software Project Management Plan
SRS	Software Requirement Specifications
UI	User Interface

1.3.3 Abbreviations

App	Application
Mgt	Management
Mtg	Meeting
Req't	Requirement
Spec	Specification

1.4 References

No references were used.

1.5 Overview of Contents of Document

This section includes the overview of each main topic of this document. The overview explains what each section includes and any other details about the sections.

1.5.1 General Description

This section contains information that makes the specific requirements (found in the next section) easier to understand. The information provided describes the users and the features that the product is to provide. In addition, this section includes any constraints that will limit the project team's options in designing the software, and any assumptions or dependencies that might affect the requirements as specified in this document.

1.5.2 Specific Requirements

This section will contain all of the technical specifications of the software including but not limited to:

- Set of all features requested by the sponsor
 - Set of all functions required by said features.
- A measurable and verifiable explanation to each feature or its functionalities
- A Traceable overview of each feature and its evolution throughout the development process

1.5.3 Appendix A - Data Dictionary

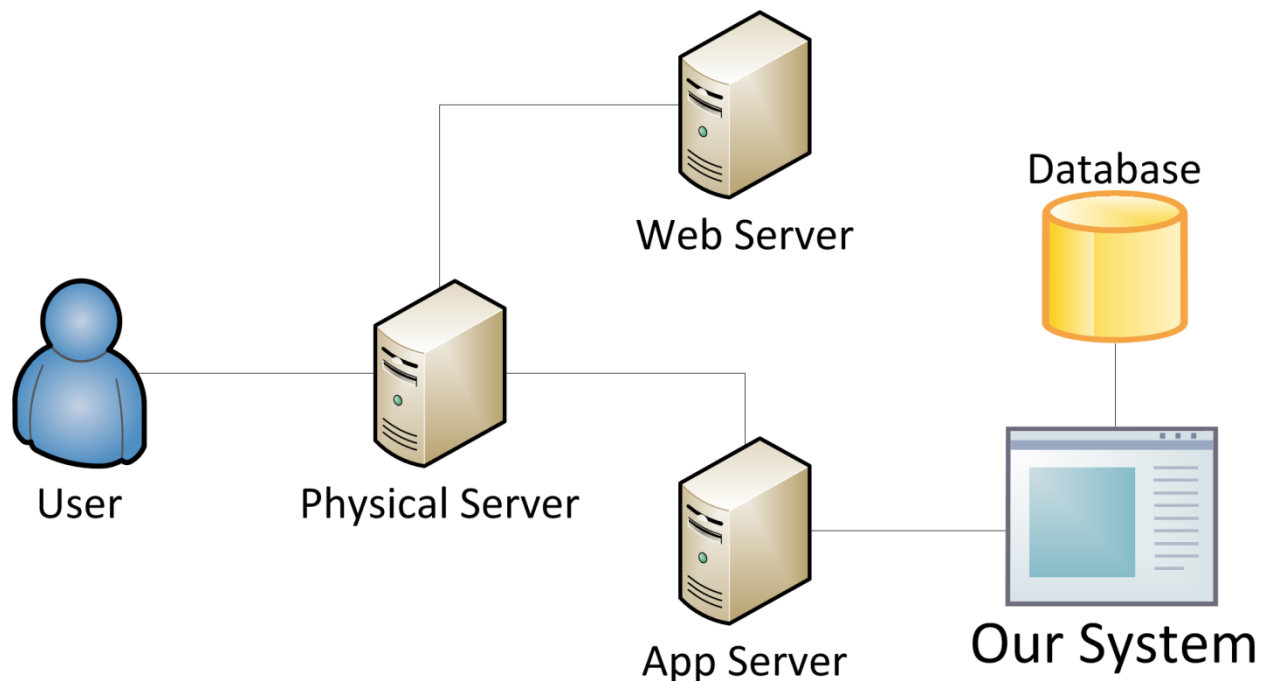
This is the Data Dictionary containing a list of all of our data entries and data tables.

2. General Description

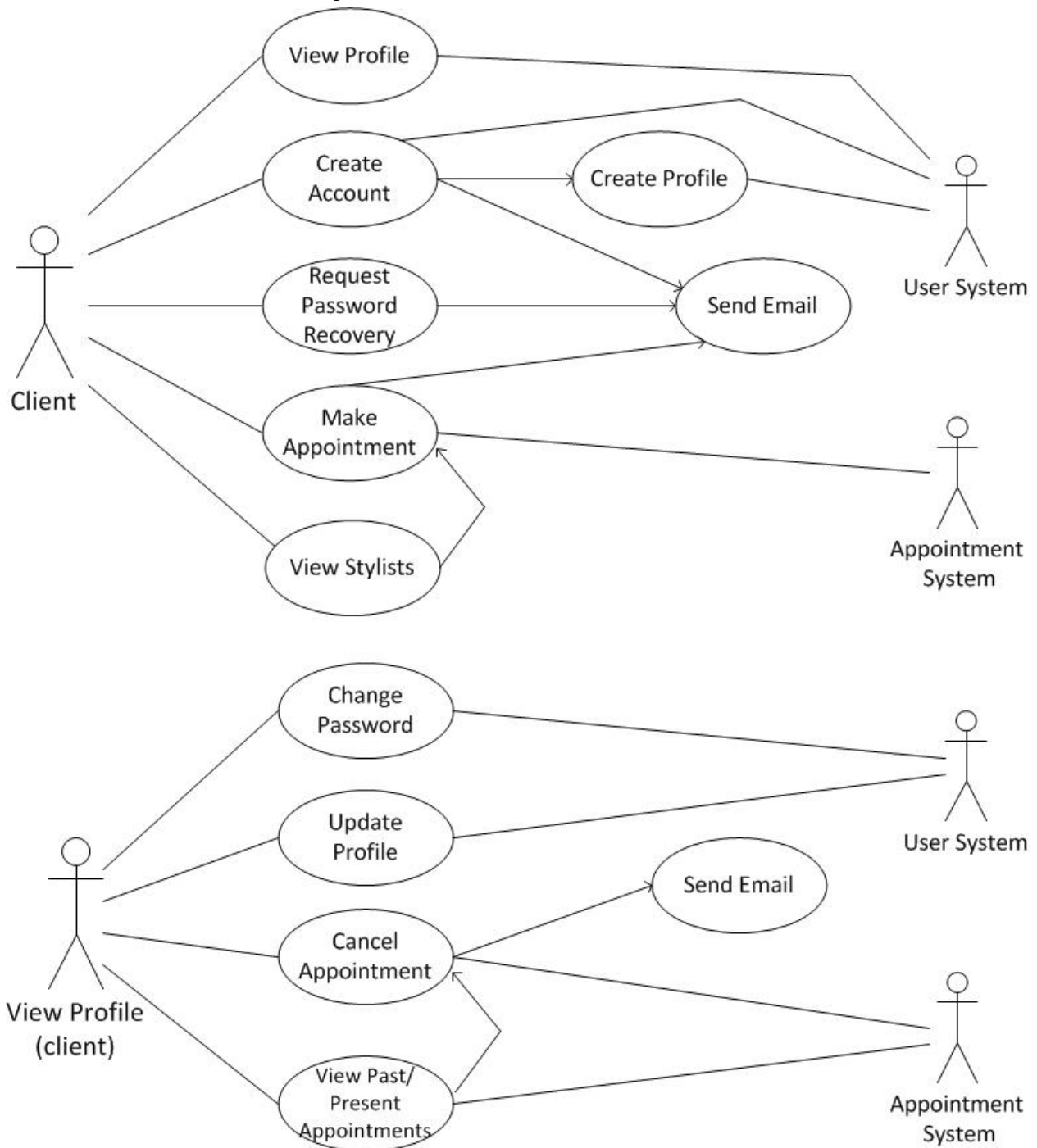
This section contains information that, hopefully, makes the specific requirements (found in the next section) easier to understand. The information provided describes the users and the features that the product is to provide. In addition, this section should include 1) any constraints that will limit the project team's options in designing the software, and 2) any assumptions or dependencies that might affect the requirements as specified in this document

2.1 Product Perspective

Our software is a web app written in Java. As such it will need an app server to facilitate its execution. Our system is independent of the salons website; however the app server that will execute our system is running on the same physical server as the website. While our system is completely independent of the sponsor's website, the website does link to our system. The following diagram is a visual representation of where our system lives in relation to the rest of the client's website:

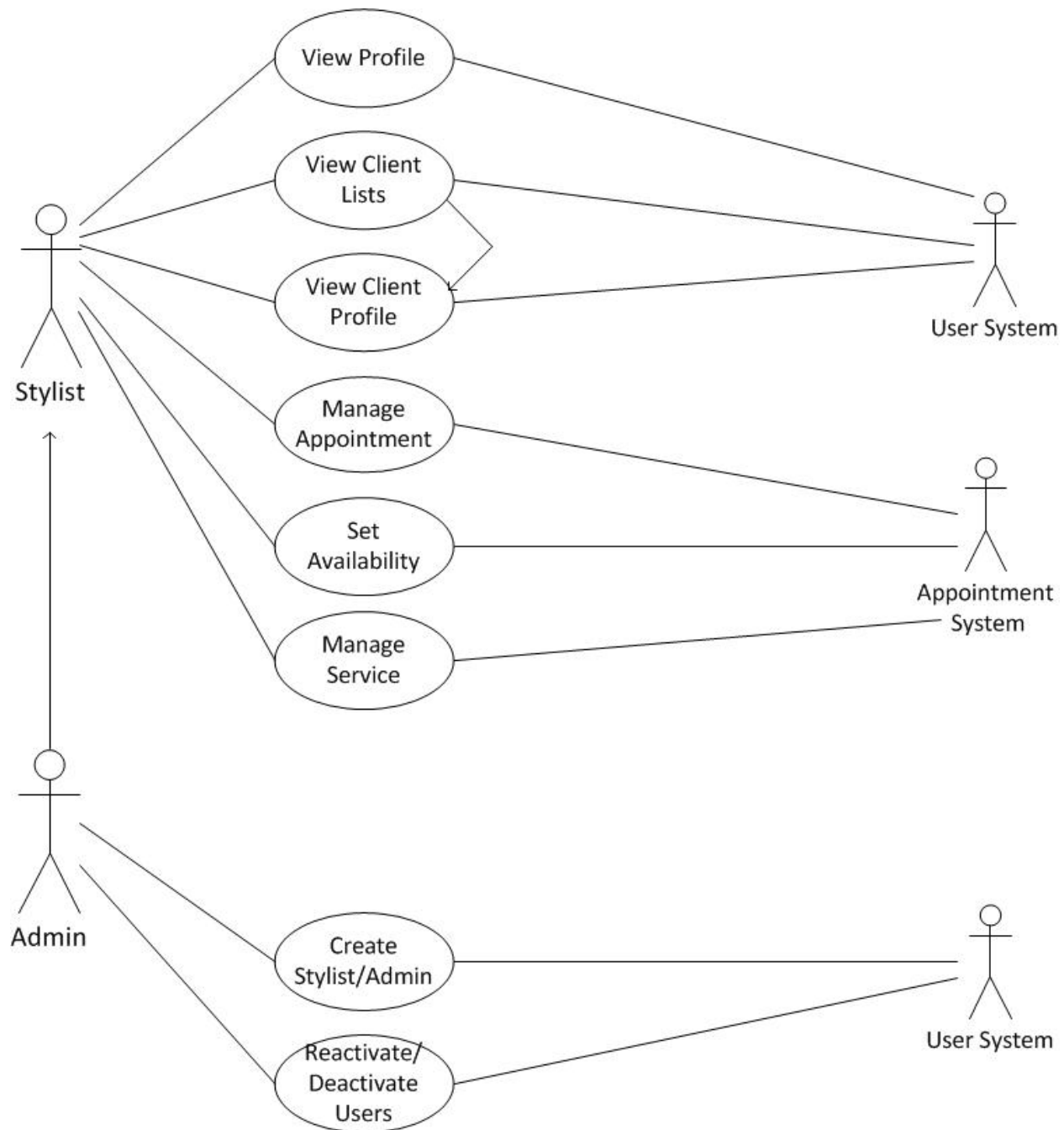


2.2 Use Case Models of System Features

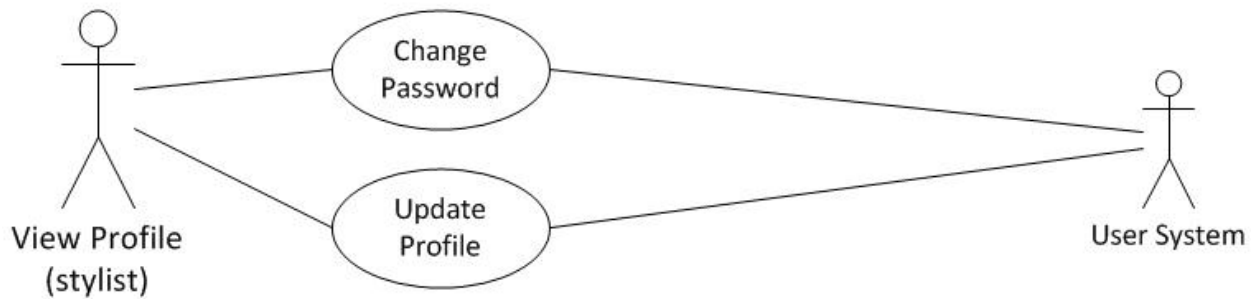
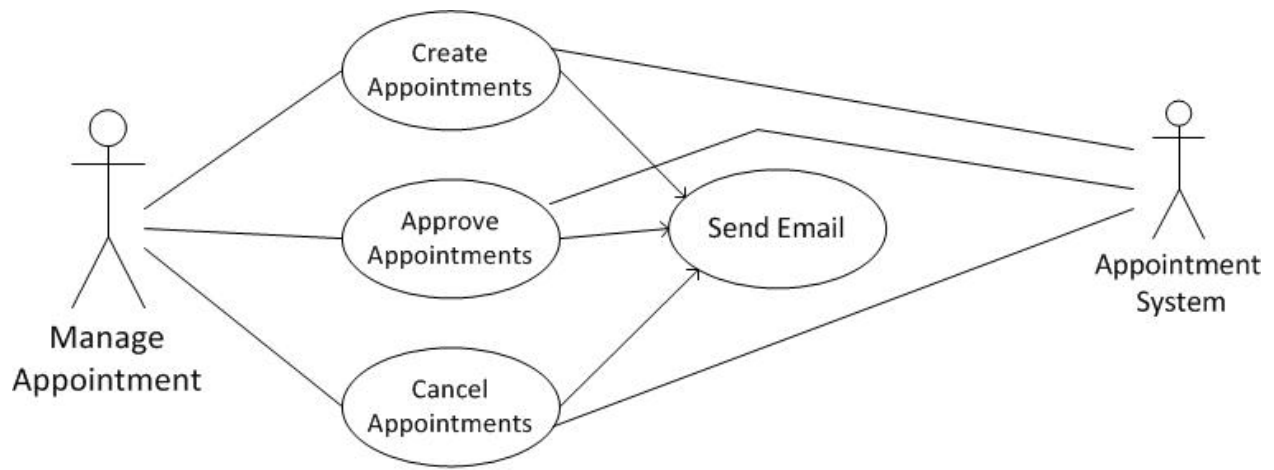


The above diagram is a high level Use Case Model for the Client actor of the system. This model shows the system features available to Clients and how those features interact with other aspects of the system that support those features.

Below the main diagram there is a further breakdown of the View Profile feature accessible by clients. This diagram is used to demonstrate how the sub-features of View Profile are used within the system. It is important to note that while this feature is similar to one of the features for Stylists and Admins, there are differences, which is why it is necessary to separate the two.



Similar to the Use Case Model for Clients, the above diagram is a high level Use Case Model for Stylist and Admin actors. This model is broken down into two sections by the type of actor, one for the Stylists and one for Admins. The model again shows the features provided by the system available to each type of actor. The Admin actor is capable of using all of the features available to stylists, as well as has access to the features noted in the model for managing the system.



These Use Case Models are a further breakdown of two of the features available to Stylists and Admins, Manage Appointment and View Profile. These breakdowns diagram related features that the high level model grouped together; this is done for both readability and completeness.

2.3 Characteristics

Users

- **Client** A client is a person who will be receiving service from our sponsor.
- **Stylist** A stylist is a service provider at our sponsor's salon.
- **Admin** An administrator user who will be in charge of maintaining and administrating users who will be using this software. An administrator will also be a stylist

Features: Client can

- Create an account
- Edit profile (Personal Information)
- Login at a later date
- Schedule an appointment
- Choose who to schedule an appointment with
- View previous appointments and appointment history
- Cancel Appointments

Features: Stylist can...

- Provide/Change availability
- Accept/Deny/Cancel appointments
- Tender an appointment
- Schedule an appointment on behalf of a client
- View previous appointments and appointment history of a given client (search)
- Cancel Appointments

Features: Admin can...

- Including all Features above
- Create and Delete users
- Set Operational Hours

2.4 General Constraints

Since our system is a web app, the system must have an app server running in order to run. We will also be storing password information which will require salting and hashing.

In order for the user to access the service, he/she will need an active internet connection and a modern browser that supports html5.

- Firefox (Mac, Windows)
- Safari (iOS, Mac, Windows)
- Chrome (iOS, Mac, Windows, Android)
- Opera (Mac, Windows)
- IE8+ (Windows)
- Dolphin (Android)

This system will rely on a Linux based app server. Based on the input from the sponsor, we have determined that the site needs to support no more than 100 live connections at a given time. In order to provide service given the above information, we have determined the following:

- Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz OR better
- Minimum 1 GB of memory
- 100 Megabit Connection
- 20 GB of Storage
- 1 TB of transfer rate per month

2.5 Assumptions and Dependencies

Throughout this documentation, we are making the following assumptions.

We are assuming that the customer will provide proper code segments so that we can associate our software into their website. The “how to” of this will not be documented until the development stage. This is due to the fact that the customer is currently in the process of creating said website.

We are assuming that the customer will utilize our system for 100% of all scheduling. If any other tools for scheduling are used, our system will not be aware of it and will allow for the booking of one time slot multiple times.

We are assuming that stylists will be 100% responsible for insuring correct input to the system when providing available times. This assumption is a result of the sponsors dynamic schedule which in real life

3. Specific Requirements

This section will contain all of the technical specifications of the software including but not limited to:

- Set of all features requested by the sponsor
 - Set of all functions required by said features.
- A measurable and verifiable explanation to each feature or its functionalities
- A Traceable overview of each feature and its evolution throughout the development process

3.1 Use Case Specifications

The following sections contain the use cases for our project. These layout the flow of the different features of our project. As the development phase continues and our clients request new features we may have to incorporate new use cases for these features.

3.1.1 Use Case 1

Use Case ID	UC-1	Use Case Name	Send Email
Created By	Mike McParland	Last Updated By	Mike McParland
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Request Send Email System		
Description	The system will give a target email, subject, and message for the system to create and send an email with. The system will then validate this information and send an email to the desired target. The email is typically a simple way to notify user about an action that the system is taking.		
Preconditions	<ol style="list-style-type: none"> 1. One of the following has occurred: new profile was created, a password recovery request, a make appointment request, or a stylists has created, approves, or cancels an appointment. 2. The system has a valid target email, subject, and message that can be used to send the Email. 3. All users have entered a valid email that the system has stored and has access too. 		
Post conditions	<ol style="list-style-type: none"> 1. Email is send to the intended user with the correct subject and message. 2. The email is archived in sent mail so it can be reviewed at a later date. 		
Normal Course	1.0 System Send Email <ol style="list-style-type: none"> 1. Request Send Email System gives email address, subject, and message. 2. System validates email that has been passed to it and that the subject and message fields are not null. 3. System creates an email with the giving inputs. 4. Email is send to desired target with correct message and subject. 5. Sent email is saved in archives. 		
Alternative Courses	None		
Exceptions	1.0.E.1 Not a valid email given (at step 2) <ol style="list-style-type: none"> 1. System displays message: Invalid email with the current email entered. 2. System will prompt to check that the user the email is being sent to a valid email. It will also give Stylist the option to edit the email address input field. <ol style="list-style-type: none"> 3a. The Stylist enter a new valid email to the intended user. 4a. The system tries to resend the email and continues on the original course. 3b. The Stylist will tell the system to cancel the email. 		
Includes	None		
Priority	High		
Frequency of Use	Used each time a password recovery is requested, an account is created, an appointment is requested, and when stylists approve, cancel, or create new appointments. It can be expected to be used at least twenty times per week.		
DB Requirements	Users		
Special Requirements	<ol style="list-style-type: none"> 1. The system has correctly set up and configured to send emails from the specified email accounts. 		
Assumptions	<ol style="list-style-type: none"> 1. The email accounts attached to the send mail system are valid. 		
Notes and Issues	None		

3.1.2 Use Case 2

Use Case ID	UC-2	Use Case Name	View Profile
Created By	Mike McParland	Last Updated By	Mike McParland
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	The Client selects to view their profile. The system loads the page that displays the current user's page. From this page the Client has several different options. On this page the client will be able to see the personal information they have enter into the system. This may include their name, email, hair color, hair style, hair length, notes, and pictures of their hair. This page also displays all the appointments that the client has scheduled for the future.		
Preconditions	<ol style="list-style-type: none"> 1. Client has created an account and entered some information. 2. Client is logged into the system. 		
Post conditions	<ol style="list-style-type: none"> 1. System displays the page of the current client's profile. 		
Normal Course	2.0 Request to View Profile <ol style="list-style-type: none"> 1. Client selects to view their profile. 2. System displays the page of the client's profile. (Alternative courses 2.1, 2.2, 2.3, 2.4) 		
Alternative Courses	2.1 Request to Change Password from View Profile (branch after step 2) <ol style="list-style-type: none"> 1. Client selects to change their password. 2. System displays the change password page. 2.2 Request to Update Profile from View Profile (branch after step 2) <ol style="list-style-type: none"> 1. Client selects to update their profile. 2. System displays the update profile page. 2.3 Request to Cancel Appointment from View Profile (branch after step 2) <ol style="list-style-type: none"> 1. Client selects to cancel a future appointment from their profile. 2. System displays the cancel appointment page. 2.4 Request to View all Appointment from View Profile (branch after step 2) <ol style="list-style-type: none"> 1. Client selects to view all appointments (past and future) from their profile. 2. System displays the page showing all the clients appointments. 		
Exceptions	2.0.E.1 Request to View Profile before creating Profile (at step 1) <ol style="list-style-type: none"> 1. System display message saying: You have not created a profile yet would you like to? <ol style="list-style-type: none"> 2a. Client selects to create their profile. 3a. System displays the page for creating a profile. 2b. Client chooses not to create their profile. 3b. System send the client back the their previous page. 		
Includes	UC-7 Change Password, UC-8 Update Profile, UC-9 Cancel Appointment, UC-10 View Past/Present Appointments		
Priority	High		
Frequency of Use	Clients will use this page whenever they want to view their profile, change their password, update profile, cancel appointments, or view their past and present appointments. Therefore they will use this several times each login.		
DB Requirements	users – userTypes, users – appointments, appointments – stylistsXAppointmentTypes, stylistsXAppointmentTypes – users, stylistsXAppointmentTypes – appointmentTypes, users – userNotes		
Special	None		

Requirements	
Assumptions	None
Notes and Issues	

3.1.3 Use Case 3

Use Case ID	UC-3	Use Case Name	Create Account
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	A new client has selected to create an account on the website. The System will display the page for creating an account and take in all the necessary inputs from the client. It will then validate all the inputs the client has entered and either creates the account sending a verification email that will activate the account and take the user to the create profile page or it will request the client enter valid information in the all the required fields.		
Preconditions	<ol style="list-style-type: none"> 1) Client does not already have a registered account under their name or email. 2) Client is on the website and wants to create a new account. 		
Post conditions	<ol style="list-style-type: none"> 1) System creates the clients account, verifies the email, and displays the create profile page for the user. 2) The clients account is stored in the User Database 		
Normal Course	3.0 Create an Account <ol style="list-style-type: none"> 1. Client selects to create an account of the website. 2. System displays the page to create an account. 3. Client enters all the required information. The required fields are the following: First Names, Last Name, and Email. 4. System validates all the information entered and uses that information to create a new user in the User Database. 5. System will use Send Email to verify the new accounts email. (alternative course 3.1) 6. Client will click a link from the email that will verify their email and activate their account on the system. 7. System will activate the clients account and take the client to the create profile page. 		
Alternative Courses	3.1 Create an Account with Invalid Email (branch after step 5) <ol style="list-style-type: none"> 1. System displays create account page again for the client asking them to please enter a new valid email address. 2. Client enters a new valid email. 3. System validates all the information entered and uses that information to create a new user in the User Database. 4. System will use Send Email to verify the new accounts email. 5. Client will click a link from the email that will verify their email and activate their account on the system. 6. System will activate the clients account and take the client to the create profile page. 		
Exceptions	3.0.E.1 Create Account with invalid inputs (at step 4) <ol style="list-style-type: none"> 1. System displays create account page again with the invalid fields mark. 2. Client enters all the required information. The required fields are the following: First Names, Last Name, Email. 3. System validates all the information entered and uses that information to create a new user in the User Database. 4. System will use Send Email to verify the new accounts email. 		

	<p>5. Client will click the link from the email that will verify their email and activate their account on the system.</p> <p>6. System will activate the clients account and take the client to the create profile page.</p>
Includes	UC-1 Send Email, UC-11 Create Profile
Priority	Low
Frequency of Use	When a client wants to create their account for the first time they must use this. Therefore each client will use this exactly once.
DB Requirements	users - userTypes
Special Requirements	None
Assumptions	None
Notes and Issues	

3.1.4 Use Case 4

Use Case ID	UC-4	Use Case Name	Request Password Recovery
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	The client wishes to recover their password because they have forgot it. The system will prompt the user to enter their email and name. The system will then use Send Email to email the client a new temporary password that they can use to login with and enter a new desired password.		
Preconditions	1) The client has a valid account already created. 2) The email account on the account is valid and the client has access to it.		
Post conditions	1) The client will be given a temporary password they can use to login with and create a new desired password. 2) New password is stored in the User Database.		
Normal Course	4.0 Request a Password Recovery <ol style="list-style-type: none"> Client selects to recover their password. System prompts the client for their name and email. Client enters their name and email and system validates inputs. System uses Send Email to Client that has a new temporary password that has been tied their account. Client uses the new password to login the system. Client selects a new desirable password using Change Password. 		
Alternative Courses	None		
Exceptions	4.0.E.1 Request a Password Recovery with invalid inputs (at step 3) <ol style="list-style-type: none"> System displays the request password recovery page again with the invalid fields mark. Client enters their name and email and system validates inputs. System uses Send Email to Client that has a new temporary password that has been tied their account. Client uses the new password to login the system. Client selects a new desirable password using Change Password. 		
Includes	UC-1 Send Email, UC-7 Change Password		
Priority	Low		
Frequency of Use	Clients will use this once or twice a year when they forget their password.		
DB Requirements	users - userTypes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.5 Use Case 5

Use Case ID	UC-5	Use Case Name	Make Appointment
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	A Client will be able to select the date of the desired appointment and the system will then display the available times of the stylist(s) on that are open on the specified date. The client will select the time of their appointment, stylist, and the services they will be getting done. The system will then create a pending appointment that will be approved by the stylist. The system will also send the stylist in question an email informing them of the appointment pending approval.		
Preconditions	1) Client is logged into system with a valid active account.		
Post conditions	1) A pending appointment request will be entered into the Appointment Database that must be approved by the stylists.		
Normal Course	5.0 Initiate Appointment Request <ol style="list-style-type: none"> Client selects the day, time, stylist, notes, and the services of their appointment. System validates the inputs. System creates a pending appointment in the appointment database that must be verified. System uses Send Email to notify the stylist they have a new pending appointment. 		
Alternative Courses	None		
Exceptions	5.0.E.1 Initiate Appointment Request with invalid inputs (at Step 2) <ol style="list-style-type: none"> System displays the Initiate Appointment Request page with the invalid fields in highlighted. Client selects the day, time, stylist, notes, and the services of their appointment. System validates the inputs. System creates a pending appointment in the appointment database that must be verified. System uses Send Email to notify the stylist they have a new pending appointment. 		
Includes	UC-1 Send Email		
Priority	High		
Frequency of Use	Clients will use this every time they want to make an appointment which ranged from once a month to once every several months.		
DB Requirements	users – appointments, appointments – stylistsXAppointments, appointments – userNotes, userNotes – users, stylistsXAppointments - users, stylistsXAppointments - appointmentTypes, stylistsXAppointments – stylistsDayOffs, stylistsXAppointments – stylistsSchedules.		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.6 Use Case 6

Use Case ID	UC-6	Use Case Name	View Stylist
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	Clients can select from the different hair stylists and nail technicians. Upon selecting one of these the system will display that stylist's or technician's profile page for the client to view. On the profile page the client will be able to see the resume, experience, techniques, and picture of the stylist or technician work. The name, email, service cost will be available on the profile page also. The client will also be able to see the available days and times that the stylist is available and can Initiate an Appointment Request with the given stylist.		
Preconditions	1) Client is logged into system with a valid active account.		
Post conditions	1) System will show the client the stylist.		
Normal Course	6.0 View Stylist <ol style="list-style-type: none"> 1. Client selects a hair stylist or nail technician from the list on the site. 2. System shows that client the stylist or technicians profile. (alternative course 6.1) 		
Alternative Courses	6.1 View Stylist with Initiate Appointment Request (branch after 2) <ol style="list-style-type: none"> 1. Client selects a date, time, and service for the current stylist or technician. 2. System uses Initiate Appointment Request with the given inputs. 3. System returns to the stylist or technicians profile once the appointment request is completed. 		
Exceptions	None (Initiate Appointment Request validates the inputs)		
Includes	UC-5 Initiate Appoint Request		
Priority	Medium-High		
Frequency of Use	Clients may use this option to schedule an appointment but there are other routes to scheduling appointments. There it will be used once every couple appointments.		
DB Requirements	stylistsXAppointmentTypes – users, stylistsXAppointmentTypes – appointmentTypes, stylistsXAppointmentTypes – stylistsSchedules, stylistsXAppointmentTypes - stylistsDayOffs		
Special Requirements	None		
Assumptions	None		
Notes and Issues	Needs review and clarity of frequency of use as well as red areas.		

3.1.7 Use Case 7

Use Case ID	UC-7	Use Case Name	Change Password
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	The client selects to change their account password from view profile. This will let them client change the password of their account by entering their old password as well as entering and then re-entering their new password. The system will then update their account to use the new password for logins.		
Preconditions	1) Client is logged into their account. 2) Client is on the view profile page.		
Post conditions	1) New password is stored in the User Database replacing the old one.		
Normal Course	7.0 Change Password <ol style="list-style-type: none"> 1. Clients select to change their password from the view profile page. 2. System shows the client the change password page. 3. Clients enter their current password once and the new desired password twice. 4. System validates the inputs of the client. 5. System changes the client's password in the User Database. 6. System returns the client to their profile page. 		
Alternative Courses	None		
Exceptions	7.0.E.0 Change Password with invalid inputs (at Step 4) <ol style="list-style-type: none"> 1. System displays the change password page that must be completely reentered. It will also display the problem with the last attempt. 2. Clients enter their current password once and the new desired password twice. 3. System validates the inputs of the client. 4. System changes the client's password in the User Database. 5. System returns the client to their profile page. 		
Includes	None		
Priority	Low		
Frequency of Use	A client may use this once or twice a year to change their password.		
DB Requirements	users - userTypes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.8 Use Case 8

Use Case ID	UC-8	Use Case Name	Update Profile
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	The client views their profile and selects to update the information. The system will then allow them to edit any of the fields on the profile page. These fields include name, email, hair color, hair style, hair length, and pictures of their hair, and notes. The system will then validate that the new updates are valid and make the changes or revert back the old changes for invalid fields.		
Preconditions	1) Client is logged into the system. 2) Client is on the view profile page.		
Post conditions	1) Updated profile information is stored in the User Database.		
Normal Course	8.0 Update Profile <ol style="list-style-type: none"> 1. Clients selects they wish to make changes to their profile. 2. System shows the client their current profile allowing them to edit the fields. (alternative course 8.1) 3. Clients enter the information they want to update. 4. System validates the inputs of the client. 5. System changes the clients profile information in the User Database. 6. System returns the client to their profile page. 		
Alternative Courses	8.1 Update Profile Cancel (branch after step 2) <ol style="list-style-type: none"> 1. Client selects to return to the view profile page. 2. System returns the client to their profile page. 		
Exceptions	8.0.E.1 Update Profile with invalid inputs (at Step 4) <ol style="list-style-type: none"> 1. System displays the update profile page with the invalid inputs highlighted. However it does make the changes to the valid inputs in the database. 2a. Client enters a valid input for the invalid fields. 3a. System validates the inputs of the client. 4a. System changes the clients profile information in the User Database. 5a. System returns the client to their profile page. 2b. Client selects to return to the view profile page. 3b. System returns the client to their profile page. 		
Includes	None		
Priority	Low		
Frequency of Use	After initially setting up the profile the client may want to update their profile from time to time but no more than five to ten times a year.		
DB Requirements	users – userTypes, users - userNotes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.9 Use Case 9

Use Case ID	UC-9	Use Case Name	Cancel Appointment
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	Clients can cancel appointments that they have made for the future with proper notice. They will be able to select which appointment they want to cancel and then confirm the cancellation. The System will then notify the Stylist via Send Email that there has been a cancellation and remove the appointment from the Appointment Database.		
Preconditions	<ol style="list-style-type: none"> 1) Client is logged in. 2) Client has appointments scheduled for the future. 3) Client is on the view profile page. 		
Post conditions	1) System removes the selected appointment from the database.		
Normal Course	9.0 Cancel Appointment <ol style="list-style-type: none"> 1. Clients select they wish to make cancel a specified appointment. 2. System shows a confirm message to the client. (alternative course 9.1) 3. Clients confirm the cancellation. 4. System notifies the stylist via Send Email of the cancellation. 5. System sets the canceled flag for the corresponding record in appointments table. 		
Alternative Courses	9.1 Cancel Appointment cancel (branch after step 2) <ol style="list-style-type: none"> 1. Clients select they do not wish to cancel the appointment. 2. System shows a confirm message to the client 		
Exceptions	None		
Includes	UC-1 Send Email		
Priority	Low		
Frequency of Use	Clients may cancel one or two appointments a year.		
DB Requirements	users – appointments		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.10 Use Case 10

Use Case ID	UC-10	Use Case Name	View Past/Present Appointment
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	Client can view their future and past appointments. The system will show a page containing the list of future and past appointments. There will be information on each appointment like the time, notes, and date. The clients can then select to cancel a future appointment if they wish.		
Preconditions	<ol style="list-style-type: none"> 1) Client has had at least one appointment in the past or has an appointment scheduled for the future. 2) Client is logged into the system. 3) Client is on the view profile page. 		
Post conditions	1) System displays all the clients past and future appointments.		
Normal Course	10.0 View Past/Present Appointments <ol style="list-style-type: none"> 1. Clients selects they wish to view past and future appointment. 2. System shows the client all their past and future appointments. (alternative course 10.1) 		
Alternative Courses	10.1 View Past/Present Appointments and Cancel Appointment <ol style="list-style-type: none"> 1. Clients selects they wish to cancel a future appointment. 2. System sends the appointment the wish to cancel to the Cancel Appointment. 		
Exceptions	none (Cancel Appointment handles validation)		
Includes	UC-9 Cancel Appointment		
Priority	Medium		
Frequency of Use	Clients will use this to view their past and future appoints so it will get used 5-10 times a year per client.		
DB Requirements	users – appointments, appointments – stylistsXAppointments, stylistsXAppointments – users, stylistsXAppointments – appointmentTypes, appointments – userNotes, userNotes - users		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.11 Use Case 11

Use Case ID	UC-11	Use Case Name	Create Profile
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Client		
Description	After creating an account the client can chose to create a profile. In this profile they can enter the following information in addition to the required name and email already entered: hair color, hair style, hair length, notes, and pictures of their hair. The system will then save this information so that the client and the stylist can view it at a later date. This information will be private from other clients.		
Preconditions	1) Client has created and activated their account. 2) Client is logged into their account.		
Post conditions	1) System puts new optional information into the User Database.		
Normal Course	11.0 Create Profile <ol style="list-style-type: none"> 1. System displays the create profile page for the client. 2. User enters in the information they want they want to have on their profile. 3. System validates the inputs the fields the client has entered. 4. System makes changes to the User Database with the client's updated information. 5. System returns client to the last page they were at. 		
Alternative Courses	None		
Exceptions	11.0.E.1 Update Profile with invalid inputs (at Step 3) <ol style="list-style-type: none"> 1. System displays the create profile page with the invalid inputs highlighted. Its saves the valid inputs and refills the valid inputs from the last attempt. 2a. Client reenters the invalid fields with valid inputs. 2b. Client leaves previously invade fields blank. 3. System validates the inputs of the client. 4. System makes changes to the User Database with the client's updated information. 5. System returns client to the last page they were at. 		
Includes	None		
Priority	Low		
Frequency of Use	Each client will use this one time.		
DB Requirements	users – userTypes, users - userTypes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.12 Use Case 12

Use Case ID	UC-12	Use Case Name	View Profile (Stylist)
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	The stylist can view their own profile. The system will load the page for the stylist that will have several options. From this page the stylist can update their profile and change their password. They can also view all their personal information, resume, services offered, and pictures of their work.		
Preconditions	<ol style="list-style-type: none"> 1. Stylist has created an account and entered some information. 2. Stylist is logged into the system. 		
Post conditions	<ol style="list-style-type: none"> 1. System displays the current stylists profile page. 		
Normal Course	12.0 Request to View Profile <ol style="list-style-type: none"> 1. Stylist selects to view their profile. 2. System displays the page of the stylist's profile. (Alternative courses 12.1, 12.2)		
Alternative Courses	12.1 Request to Change Password from View Profile (branch after step 2) <ol style="list-style-type: none"> 1. Stylist selects to change their password. 2. System displays the change password page. 12.2 Request to Update Profile from View Profile (branch after step 2) <ol style="list-style-type: none"> 1. Stylist selects to update their profile. 2. System displays the update profile page. 		
Exceptions	12.0.E.1 Request to View Profile before creating Profile (at step 1) <ol style="list-style-type: none"> 1. System display message saying: You have not created a profile yet, would you like to? 2a. Stylist selects to create their profile. 3a. System displays the page for creating a profile. 2b. Stylist chooses not to create their profile. 3b. System send the Stylist back the their previous page. 		
Includes	UC-18 Password Change, UC-19 Update Profile		
Priority	High		
Frequency of Use	Each stylist will use this five times a day at a week.		
DB Requirements	users – stylistsXAppointments, stylistsXAppointments - appointmentTypes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.13 Use Case 13

Use Case ID	UC-13	Use Case Name	View Client List
Created By	Kyle Matz	Last Updated By	Kyle Matz
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	The Stylist can select to view all their clients in the system. This list is set up so that each client name is a link to that client's profile page. The admin stylists will be able to view all the different stylist's clients but normal stylists will only be able to view their own clients. They stylists can select on a particulate client to see more information and that clients profile page.		
Preconditions	1) The stylist has a valid account. 2) The stylist is logged into their account.		
Post conditions	1. Displays a list of clients.		
Normal Course	13.0 View Client List 1. From their profile page a Stylist or Admin will select to view client list. 2. The system displays a page with a list of all the past and present clients of the stylist. (alternative course 13.1)		
Alternative Courses	13.1 View Client's Profile from Client List (branch after 2) 1. The User clicks on a client's entry in the list. 2. The system displays the client's full profile page.		
Exceptions	13.0.E.0 View List of Associated Clients When None Exist (at step 1) 1. System displays the message: "You do not have any associated clients." This message is displayed where the list would usually be.		
Includes	UC-14 View Client Profile		
Priority	Medium-High		
Frequency of Use	Due to the view client list leading to the client's profile it will be used somewhat frequently.		
DB Requirements	users – userTypes, users – stylistsXAppointmentTypes, stylistsXAppointmentTypes – appointments, appointments - users		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.14 Use Case 14

Use Case ID	UC-14	Use Case Name	View Client Profile
Created By	Kyle Matz	Last Updated By	Mike McParland
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	The Stylist selects a client whose profile they wish to view. The profile will contain all of the personal information the client has entered into the system with an additional client notes text area hidden from the clients view.		
Preconditions	1) The stylist is logged in with active account. 2) The stylist is on the view client list page.		
Post conditions	1. The system displays the clients full profile page of the selected client.		
Normal Course	14.0 View Client Profile <ol style="list-style-type: none"> 1. From the view client list page the stylist will select a client to view their profile. 2. System will display the full profile for the client selected including an editable notes text area. (alternative course 14.1) 		
Alternative Courses	14.1 Change Hidden Client Note (branch after step 2) <ol style="list-style-type: none"> 1. The stylist or admin types in the client notes text area and selects to saves the changes. 2. The system changes the note area of the clients profile in the User Database. 		
Exceptions	None (valid client ensured by client list)		
Includes	None		
Priority	High		
Frequency of Use	The client's profile page will contain valuable information for Stylists and could be visited frequently to view this information.		
DB Requirements	users - userNotes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.15 Use Case 15

Use Case ID	UC-15	Use Case Name	Manage Appointments
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	Stylists will be able to create new appointments, manage pending appointments, and cancel future appointments. The system will also list all the future appointment on top in order for ease of access. The past appointment will be on the bottom so they can be viewed also. The stylists will also be able to view a list of appointments based off of a time and date ranged entered.		
Preconditions	1) Stylist must be logged in to an active account.		
Post conditions	1) System shows the stylist the manage appointment pages.		
Normal Course	15.0 Manage Appointments <ol style="list-style-type: none"> 1. Stylist select to view the manage profile page. 2. System will display the manage profile page for the stylist. This page will list the appointments with a bit of information such as the appointment notes. (alternative course 15.1, 15.2, 15.3)		
Alternative Courses	15.0 Create Appointment from Manage Appointments (branch after step 2) <ol style="list-style-type: none"> 1. Stylist will select to create a new appointment from the manage appointments page. 2. System will display the Create Appointments page. 15.1 Manage Pending Appointments from Manage Appointments <ol style="list-style-type: none"> 1. Stylist will select to manage pending appointments from the manage appointments page. 2. System will display the Manage Pending Appointments page. 15.2 Cancel Appointment from Manage Appointments <ol style="list-style-type: none"> 1. Stylist will select to cancel a future appointments from the manage appointment page. 2. System will display the Manage Pending Appointments page. 		
Exceptions	15.0.E.1 Request to Manage Appointments before having an appointments (at step 1) <ol style="list-style-type: none"> 1. System will display the Manage Appointments page but will display that the stylist has no past or future appointments where the appointments would be shown. 		
Includes	UC-20 Create Appointments, UC-21 Managing Pending Appointments, UC-22 Cancel Appointment		
Priority	High		
Frequency of Use	Anytime a stylist wants to manage appointments they must use this page. Therefore it will be used several times a day by each stylist.		
DB Requirements	users – stylistsXAppointmentTypes, stylistsXAppointmentTypes – appointmentTypes, stylistsXAppointmentTypes – appointments, appointments – users, appointments - userNotes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.16 Use Case 16

Use Case ID	UC-16	Use Case Name	Set Availability
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	The stylist will be able to set their available hours for each day of the week. These hours will be stored by the system and then used to generate the stylist's availability schedule. They stylist can change their hours at will and the system will update and reflect these changes. If the change in hours affects a future appointment the system will pop up a message informing the stylist that an appointment is now scheduled outside of their availability. They stylist can either select to keep the appointment or reschedule the appointment.		
Preconditions	1) Stylist is logged in to an activated account.		
Post conditions	1) Stylist can view and edit their current availability. 2) System will update their current stylist's availability.		
Normal Course	16.0 Set Availability <ol style="list-style-type: none"> 1. Stylist selects to view or edit their availability. 2. System shows the stylists their current availability. (alternative course 16.1) 3. Stylist enters the new availability they want the system to use. 4. System validates the new availability. 5. System updates the availability of the stylist and updates their current schedule on the system. (alternative course 16.2, 16.3) 		
Alternative Courses	16.1 Set Availability no change (branch after step 2) <ol style="list-style-type: none"> 1. Stylist decided their availability is currently fine and do not need to change it. 2. System shows the stylist the last page they were at. 16.2 Set Availability with conflict (branch after step 5) <ol style="list-style-type: none"> 1. System displays message informing the stylist that their change of availability conflicts with a future scheduled appointment. 2. The stylists select to keep they appointment even though it is outside their availability. 16.3 Set Availability with conflict and reschedule (branch after step 5) <ol style="list-style-type: none"> 1. System displays message informing the stylist that their change of availability conflicts with a future scheduled appointment. 2. The stylists selects they wish the cancel the appointment and the appointment is sent to Cancel Appointment. 		
Exceptions	16.0.E.1 Set Availability with invalid inputs (at step 4) <ol style="list-style-type: none"> 1. System will display a message informing the stylist that their inputted fields for the availability are invalid. 2. System will prompt the stylists to enter valid inputs. 3. System validates the new availability. 4. System updates the availability of the stylist and updates their current schedule on the system. 		
Includes	UC-22 Cancel Appointments		
Priority	Medium		
Frequency of Use	Each stylist will use this to change their availability a couple times a month.		
DB Requirements	users – stylistsSchedules, stylistsSchedules – days, users – stylistsDayOffs		
Special	None		

Requirements	
Assumptions	None
Notes and Issues	None

3.1.17 Use Case 17

Use Case ID	UC-17	Use Case Name	Manage Service
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	Stylists can add, update, or removed the services they offer on the site. The system will display all the services and information with each service so that the stylists can edit, remove, or add a new service. There services will include a name, estimated cost, and estimated appointment duration.		
Preconditions	1) Stylist is logged in to an activated account.		
Post conditions	1) System will display the service the stylist offers and allow then to update, edit, or remove one. 2) The system will make the appropriate changes in the database.		
Normal Course	17.0 Manage Service <ol style="list-style-type: none"> 1. Stylist will select to manage their services. 2. System will display the manage service page for the stylist. (alternative course 17.1, 17.2, 17.3) 3. Stylist select not to edit, add, or remove any services. 		
Alternative Courses	17.1 Manage Service adding service (branch after step 2) <ol style="list-style-type: none"> 1. Stylist will select to add a new service. 2. System will display page to add new service. 3. Stylist will input service name, cost, and timeframe. 4. System will validate the inputs. 5. System will add the new service to the current stylist services. 17.2 Manage Service editing service (branch after step 2) <ol style="list-style-type: none"> 1. Stylist will select to edit an existing service. 2. System will display the service they wish to update. 3. Stylist will edit the service name, cost, and timeframe if desired. 4. System will validate the inputs. 5. System will edit the service in the database. 17.3 Manage Service removing service (branch after step 2) <ol style="list-style-type: none"> 1. Stylist will select to remove a service. 2. System displays a confirm page with the service they wish to delete. <ol style="list-style-type: none"> 3a. Stylist will select to confirm deletion. 4a. System will removed the select service from the database. 3b. Stylist will select to cancel the removal. 4b. System will return the stylists to the manage service. 		
Exceptions	17.1.E.1 Add service with invalid inputs (at step 4) <ol style="list-style-type: none"> 1. System will display a message informing the stylist that their inputted fields for the availability are invalid. 2. System will prompt the stylists to enter valid inputs. 3. System validates the new inputs for a service. 4. System adds the new service for the stylist in the database. 17.2.E.1 Update service with invalid inputs (at step 4) <ol style="list-style-type: none"> 1. System will display a message informing the stylist that their inputted fields for the availability are invalid. 		

	<ol style="list-style-type: none"> 2. System will prompt the stylists to enter valid inputs. 3. System validates the updated service information. 4. System edits the service for the stylist in the database.
Includes	None
Priority	Low
Frequency of Use	Each stylist will use this once or twice a year to update their available services.
DB Requirements	users – stylistsXAppointmentTypes, stylistsXAppointmentTypes – appointmentTypes,
Special Requirements	None
Assumptions	None
Notes and Issues	None

3.1.18 Use Case 18

Use Case ID	UC-18	Use Case Name	Password Change
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	The stylist selects to change the password on their account. The system will prompt the user for their old password and then to enter the new desired password twice. The system will then validate and either asks for a valid input or change the password into the user database.		
Preconditions	1) Stylist is logged into their account.		
Post conditions	1) System will store new password in the user database replacing the old one.		
Normal Course	18.0 Change Password <ol style="list-style-type: none"> 1. Stylists select to change their password from the profile page. 2. System shows the stylists the change password page. 3. Stylists enter their current password once and the new desired password twice. 4. System validates the inputs of the stylist. 5. System changes the stylist's password in the User Database. 6. System returns the stylists to their profile page. 		
Alternative Courses	None		
Exceptions	18.0.E.1 Change Password with invalid inputs (at Step 4) <ol style="list-style-type: none"> 1. System displays the change password page that must be completely reentered. It will also display the problem with the last attempt. 2. Stylist enters their current password once and the new desired password twice. 3. System validates the inputs of the stylist. 4. System changes the stylist's password in the User Database. 5. System returns the stylist to their profile page. 		
Includes	None		
Priority	Low		
Frequency of Use	Each stylist will use this feature maybe a couple times a year.		
DB Requirements	Users		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.19 Use Case 19

Use Case ID	UC-19	Use Case Name	Update Profile
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	The stylist views their profile and selects to update the information on their profile page. The system will then allow them to edit any of the fields on the profile page. These fields include name, email, resume, qualifications, examples of past work, and additional notes.		
Preconditions	1. The stylist is logged into the system with valid account. 2. The stylist is on their profile page.		
Post conditions	1. Updated profile information is stored in the User Database.		
Normal Course	19.0 Update Profile <ol style="list-style-type: none"> 1. Stylists selects they wish to make changes to their profile. 2. System shows the stylist their current profile allowing them to edit the fields. (alternative course 19.1) 3. Stylists enter the information they want to update. 4. System validates the inputs of the stylist. 5. System changes the stylists profile information in the User Database. 6. System returns the stylists to their profile page. 		
Alternative Courses	19.1 Update Profile Cancel (branch after step 2) <ol style="list-style-type: none"> 1. Stylist selects to return to the view profile page. 2. System returns the stylist to their profile page. 		
Exceptions	19.0.E.1 Update Profile with invalid inputs (at Step 4) <ol style="list-style-type: none"> 1. System displays update profile page with the invalid fields highlighted and the valid fields updated in the database. 2a. Stylist re-enters invalid inputs and submits changes again to system. 3a. System validates the inputs of the stylist. 4a. System changes the stylists profile information in the User Database. 5a. System returns the stylists to their profile page. 2b. Stylist decides not to re-enter the invalid fields and is happy with the changes. 3b. System returns to the stylist profile page. 		
Includes	None		
Priority	Medium		
Frequency of Use	Each stylist will use this every time they want to update their profiles information.		
DB Requirements	users - userNotes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.20 Use Case 20

Use Case ID	UC-20	Use Case Name	Create Appointment
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	The stylist can select to create an appointment for a given client. The system will then send an email notifying the client that the appointment is scheduled. This appointment does not need to be approved by the stylist as they are the one making it. Once submitted it will be added to the future appointments of the stylist in the database.		
Preconditions	1) Stylist is logged in an active account. 2) Stylist is on the manage appointments page.		
Post conditions	1) System uses Send Mail to notify client of the appointment. 2) System creates an appointment in the database.		
Normal Course	20.0 Create Appointment <ol style="list-style-type: none"> 1. Stylist selects to create an new appointment. 2. System shows the create appointment page. 3. Stylist enter the client name, appointment date, appointment time, and service selected 4. System validates the inputs of the stylists. 5. System makes the appointment in the appointment database. 6. System uses Send Email to notify the client of the new appointment. 7. System returns the stylist to the manage appointments page. 		
Alternative Courses	None		
Exceptions	20.0.E.1 Create Appointment with invalid (at Step 4) <ol style="list-style-type: none"> 1. System displays crate appointment page again with the invalid fields highlighted. 2. Stylist re-enters invalid inputs with valid responses and submits create appointment again. 3. System validates the inputs of the stylist. 4. System makes the appointment in the appointment database. 5. System uses Send Email to notify the client of the new appointment. 6. System returns the stylist to the manage appointments page. 		
Includes	UC-1 Send Email		
Priority	Medium		
Frequency of Use	Stylist will use this whenever they need to make an appointment for a given client. Most of the time clients will request to make their own appointments and therefore this will only be used a couple times a year.		
DB Requirements	users – stylistsXAppointmentTypes, stylistsXAppointmentTypes – appointmentTypes, stylistsXAppointmentTypes – appointment, appointments – users, appointments – userNotes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.21 Use Case 21

Use Case ID	UC-21	Use Case Name	Manage Pending Appointment
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	Stylists can view all their pending appointments on the system. The stylist can they view the appointment details and either approve the appointment or cancel the appointment for the client. The system will send an email the client to inform then whether or not the appointment has been approved. If approved the system will set the pending appointment as an approved appointment.		
Preconditions	<ol style="list-style-type: none"> 1) Stylist is logged into a valid account. 2) Stylist on the Manage Appointment page. 		
Post conditions	<ol style="list-style-type: none"> 1) System either approves or denies the pending appointment. 2) System email the client whether or not their appointment is approves. 3) System will mark the appointment as approved or denied. 		
Normal Course	21.0 Manage Pending Appointment <ol style="list-style-type: none"> 1. Stylist selects to manage pending appointment. 2. System shows the pending appointments page. (alternative course 21.1) 3. Stylist select the appointment they wish to approve and can review the details before submitting the approval to the system. 4. System marks the appointment as approved in the database. 5. System emails the client that their appointment is approved. 		
Alternative Courses	21.1 Manage Pending Appointment Deny (branch after step 2) <ol style="list-style-type: none"> 1. Stylist selects the appointment they wish to deny and any note they want the system to include with the email informing the client of the appointment denial. 2. System validates inputs 3. System marks the appointment as a cancelled appointment. 4. System uses Send Email to send the client the email. 		
Exceptions	21.1.E.1 Manage Pending Appointment Deny invalid note input (at Step 2) <ol style="list-style-type: none"> 1. System displays the invalid note the stylist entered and asked for a valid input for that field. 2. Stylist re-enters valid inputs and submits deny appointment again. 3. System validates the inputs of the stylist. 4. System marks the appointment as a cancelled appointment 5. System uses Send Email to send the client the email. 		
Includes	UC-1 Send Email		
Priority	High		
Frequency of Use	Each Stylist will use this feature every appointment that a client schedules. This could be up to several times a week.		
DB Requirements	users – stylistsXAppointmentTypes, stylistsXAppointments – appointmentTypes, stylistsXAppointmentTypes – appointments, appointments – users, appointments – userNotes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.22 Use Case 22

Use Case ID	UC-22	Use Case Name	Cancel Appointment
Created By	Mike McParland	Last Updated By	Mike McParland
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Stylist		
Description	A stylist can cancel any upcoming appointments. They will be able to select which appointment they want to cancel and then confirm the cancelation. The system will then notify the client via email that the cancelation has been made and remove the appointment from the Appointment Database.		
Preconditions	<ol style="list-style-type: none"> 1. Stylist or Admin is logged into the system. 2. Stylist or Admin has an appointment scheduled for the future. 3. Stylist or Admin is on view profile page or manage appointments page. 		
Post conditions	<ol style="list-style-type: none"> 1. System removes the appointment from the database. 2. System emails the client 		
Normal Course	22.0 Cancel Appointment <ol style="list-style-type: none"> 1. Stylist selects they wish to cancel a specified appointment. 2. System shows a confirmation message to the user. (alternative course 22.1) 3. Stylist confirms the cancelation. 4. System notifies the client via Send Email of the cancelation. 5. System removes the appointment from the Appointment Database. 		
Alternative Courses	22.1 Cancel Appointment Cancel (branch after step 2) <ol style="list-style-type: none"> 1. Stylist select they do not wish to cancel the appointment. 2. System shows confirmation message to the user. 		
Exceptions	None		
Includes	UC-1 Send Email		
Priority	Low		
Frequency of Use	Stylists may cancel one or two appointments a year.		
DB Requirements	users – stylistsXAppointmentTypes, stylistsXAppointments – appointmentTypes, stylistsXAppointmentTypes – appointments, appointments – users, appointments – userNotes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.23 Use Case 23

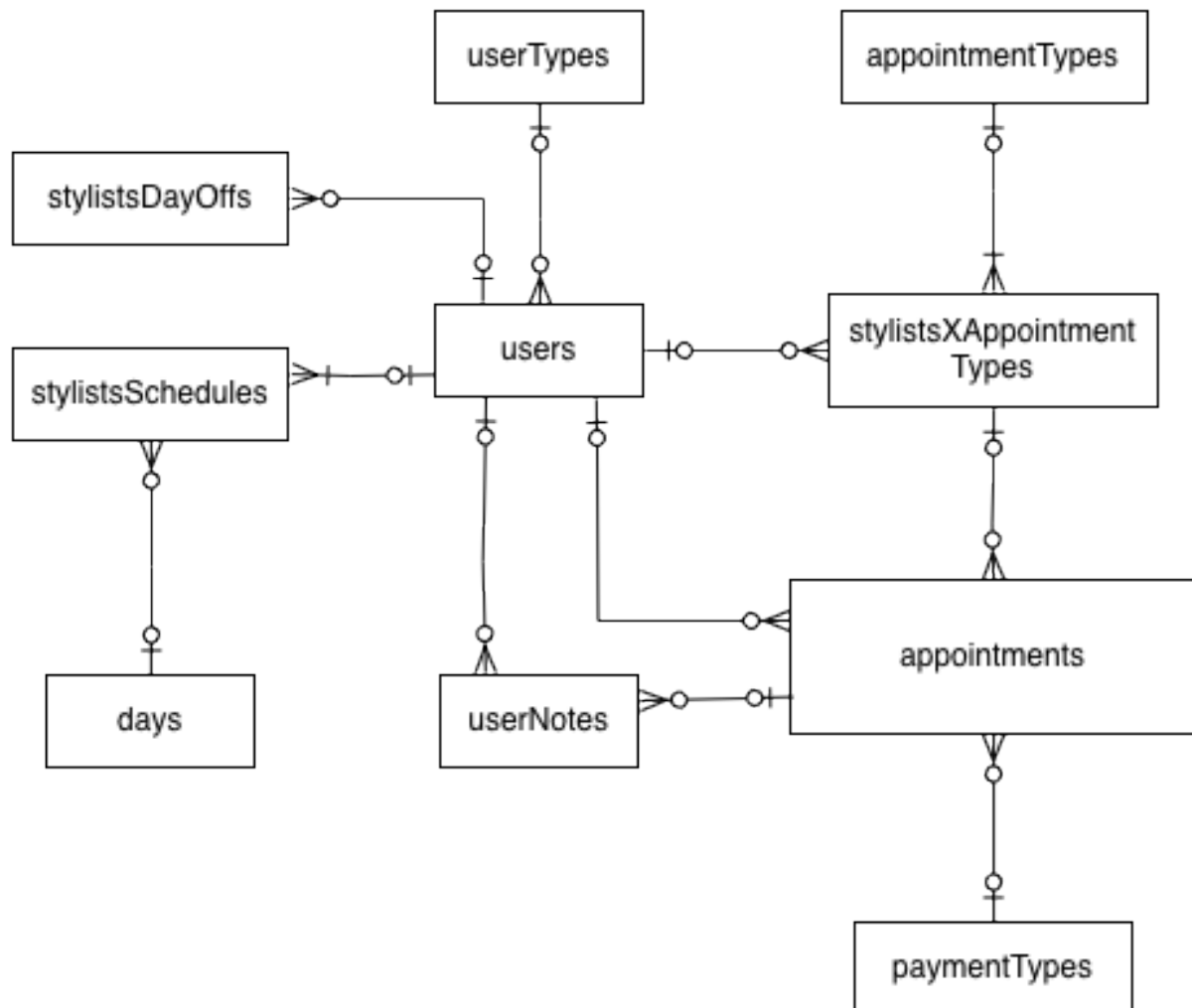
Use Case ID	UC-23	Use Case Name	Create Stylist/Admin
Created By	Mike McParland	Last Updated By	
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Admin		
Description	The system Admin can create an account for another admin or a stylist. The admin will enter the name, email, account type (admin or stylist), and a password for the new account. The system will then validate these fields and create a new user in the database. The system will also email the new account informing them it has been created though the new user does not need to take action to activate the account.		
Preconditions	<ol style="list-style-type: none"> 1. Admin is logged in with an active account. 2. Admin is on the create new account page. 		
Post conditions	<ol style="list-style-type: none"> 1. System creates a new account for the new stylist or admin in the database. 2. System emails new account that it has been created. 		
Normal Course	23.0 Create new Stylist/Admin <ol style="list-style-type: none"> 1. Admin selects to create a new stylist or admin. 2. System displays the create stylist/admin page. 3. Admin enters the name, email, account type, and password. 4. System validates the inputs. 5. System creates a new account in the database. 6. System uses Send Email to inform the new user of the account creation. 		
Alternative Courses	None		
Exceptions	23.0.E.1 Create new Stylist/Admin with invalid inputs (at step 4) <ol style="list-style-type: none"> 1. System display create new stylist or admin page with invalid inputs highlighted. 2. Stylist re-enters inputs with valid fields. 3. System validates the new inputs. 4. System creates a new account in the database. 5. System uses Send Email to inform the new user of the account creation 		
Includes	UC-1 Send Email		
Priority	High		
Frequency of Use	Each time the admin needs to create a new admin or stylist so once or twice a year.		
DB Requirements	users - userTypes		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.24 Use Case 24

Use Case ID	UC-24	Use Case Name	Reactivate/Deactivate User
Created By	Kyle Matz	Last Updated By	Kyle Matz
Date Created	4/20/13	Date Last Updated	4/25/13
Actors	Admin		
Description	The administrators can reactivate or deactivate any other type of account other than their own. The system will list all the current accounts and the admin can select which account they wish to reactivate or deactivate. This will make the system mark the account as inactive and unable to login or activate and able to log in again. Any account made deactivate will still be stored in the database but will unable to login.		
Preconditions	1) The admin is logged in on an active account. 2) There are other accounts in the system to be deactivated.		
Post conditions	1) Account is marked as deactivated in the database.		
Normal Course	24.0 Deactivate User <ol style="list-style-type: none"> Admin will select the reactivate or deactivate an account. The system will display all the activate account on top and all the de-active accounts on the bottom. (alternative course 13.1) Admin will select the account they wish to deactivate. System will mark the account as de-active. 		
Alternative Courses	24.1 Reactivate User (branch after 2) <ol style="list-style-type: none"> Admin will select a de-active account the wish to reactivate. System will mark the account as active. 		
Exceptions	None		
Includes	None		
Priority	Low		
Frequency of Use	Administrators may deactivate or reactivate one or two accounts per year.		
DB Requirements	users		
Special Requirements	None		
Assumptions	None		
Notes and Issues	None		

3.1.25 ERD

The ERD below contains all of the entities and the relationships required to support the Use Cases described previously. These specifications will be used to design the relational database to be developed as part of the software system. The users - userTypes and users - userNotes relationships will provide necessary details for the implementation of the Client Management System which will encompass account and profile creation and modification. users - stylistsXAppointmentTypes, stylistsXAppointmentTypes - appointmentTypes relationships will allow stylists to assign service types to themselves. users - stylistsSchedules and users - stylistsDayOffs will allow stylists to set their days and times of availability. The users - appointments and appointments - stylistsXAppointments relationships will provide Appointment Management system functionality for both clients and stylists.



3.2 Performance Requirements

Based on our sponsor's needs, we have determined that only one server is needed that will support a virtually unlimited number of connections for the clients (users) to use.

Our software does not limit the number of active users; however, based on our sponsor's needs we have limited the hardware that will be supporting our software. Based on that, our application will support up to 100 active connections.

3.3 Design Constraints

The only constraint the sponsor has for our system is its hosting costs. They have stated they do not wish to spend a lot of money on hosting. Due to this we know that the system running our app will not be extremely powerful. All of the options they can choose from support MySQL for a database and have at least 20GB of hard disk space.

3.4 Quality Attributes

This subsection contains the non-functional requirements, that is, requirements that apply to the software system being developed and not just to a single function or use.

3.4.1 Reliability

Our system will perform at 99.9% uptime given the sponsors conditions and explanations of the user base. The 0.1% downtime is an operational constraint due to Linode promising 99.9% active uptime. This may change based on the app server being used.

3.4.2 Maintainability

The system will be easily maintainable for Java developers. Code will be well commented to allow for quick comprehension. The dynamic nature of the environment will allow for easy movability if needed during extended down time of the server. The code will also be commented using Javadoc which will generate html web pages that can be used to easily maintain the code.

3.4.3 Program Quality Attributes

- Readability: We will ensure easy readability of the code. This is enforced by the agile approach that our team has taken towards the development of this software through all of its stages. Through agile development, all members of the team have to be able to read all of the code and understand it. We have also agreed upon the following coding standard.
 - All functions will have descriptive function name
 - e.g.: a string manipulation function that takes in a string and outputs the string into upper cases will be name in the following manner
 - `changeStringToUpperCase()`
 - Javadocs style commenting for everything
 - This will allow us to export our comments and easily make a manual for the people who will be maintaining the code.
- Robustness: Our software will handle inputs in one of two ways:
 - Through UI: Our interface will not allow for incorrect input.
 - e.g.: If a given field requires a phone number a submit button will not be active (clickable) unless the phone number is in the correct format
 - Server side sanitation: After a given input is submitted to the server, the application will sanitize all inputs to ensure that no malicious inputs have been bypassed through various means.
 - The above methods will ensure the security of the server and ease of use for all users of the application
- Speed of Execution: Using the Spring framework, we can ensure fast code execution. As developers, our code will be turned into objects and given to the JVM. We will use Java modules to handle database communication as well as computations which have been optimized by the open source community.

3.4.4 Security

Unauthorized use of the software:

- The system will not allow unregistered users access to any part of the system.
- The system will store user information such as emails, phone numbers and names. To ensure security and privacy of the passwords, they will be salted and hashed using a cryptographic hash.
- Furthermore, we will utilize proper sanitization to protect against SQL injections. The server itself will only allow HTTP(S) traffic from all IP addresses and SSH access only from the owner of the server for further administration. In return, this will disable attack vectors to services that are not implemented. ie: No SMTP connections to the host server is needed or required by our application.
- Due to the restrictive nature of the host server, no viruses can penetrate or remotely install on the server

Accidental data loss:

- The administrator of the website is responsible for daily, weekly and bi weekly backups of the database. In the event of data loss in the database, an administrator of the server can resume to an earlier backup minimizing data loss.

3.4.5 Transferability / Conversion

Our system will be written in Java and will be executed by an app server such as Tomcat or Resin. Because of Java's cross platform nature, the software can run on any computer that has an app server and the JDK installed. Furthermore, the end software once compiled from source, is a executable java file which can be inserted to any java supported app server and can be live within seconds. The database can also be exported and imported using various open source MySQL user interfaces.

3.4.6 Operational Quality Attribute

- Ergonomic aspects
 - High Contrast: The contrast between the information displayed on the page and the background of the page will be high enough so that there is no visibility issues for a user.
- Screen layouts
 - Browser window with at least a width 960 pixels.
- User interface languages
 - English: The app will be purely based on the English language. The salon is an English based business and the web app will reflect this aspect.

3.4.7 Operations

Our system only has one mode of operation, running. There are no special forms of operation related to backing up data, updating the system or any other tasks related to the system.

3.4.8 Site Adaptations

The system is a Java web app. The system needs to be installed on a computer running an app server. The installer must install and setup the app server on the computer hosting the system. He must also install the Java 7 JDK. Once both of those have been installed, the installer must place our java executable into the app servers app directory. Once there, the system will be run automatically every time a user visits the server from a web browser.

4. Approvals

Project Sponsors

By signing this Software Requirements Specification document you agree that this document represents a shared understanding that the requirements for the project and the system described will satisfy your needs. Any future changes in this baseline specification will be made through the project's defined change process. You understand that approved changes might require the renegotiation of the scope of the work that can be completed within the allotted time. Approval also constitutes an agreement between you and the development team that the requirements, as stated, provide all that is necessary for the development team to proceed with the design and implementation of the software system.

Lisa Sigurdson _____

Alayna Sigurdson _____

Advisor

By signing this Software Requirements Specification document you are approving the document meets necessary standards for this CSc 190 class.

Ahmed M Salem _____

Team Sierra

By signing this Software Requirements Specification document you agree that all requirements listed in this document will be met for the final project unless a change is agreed upon by both you and the project sponsor.

Alex Chernyak _____

Joubin Jabbari _____

Kyle Matz _____

Mike McParland _____

Scott Livingston _____

Serge Lysak _____

Appendix A - Data Dictionary

This is the Data Dictionary containing a list of all of our data entries and data tables.

Data Element

A list of all the different data elements used in our database.

Element	Meta
amountPaid	<ul style="list-style-type: none">• Description: An element that will hold the estimated cost per service• Type: DECIMAL (5, 2)• How it's used: Used to store the final cost of the specified appointment.• How its set: The stylist will enter the total cost of the appointment once completed.• Validation: The application will require a valid decimal (5,2) or prompt for the user to reenter the value.
appointmentID	<ul style="list-style-type: none">• Description: This is be an integer that will store the appointment number.• Type: INT• How it's used: This will be used to keep track of the different appointments. Each appointment will have a unique integer value.• How its set: Each ID will be created by the system at the appointment generation.
appointmentTypeID	<ul style="list-style-type: none">• Description: A unique integer value that links an appointment with the appointment types table.• Type: INT• How its used: Used to link the appointments and appointment types tables.• How its set: The application generates this value when an appointment is created.

clientID	<ul style="list-style-type: none"> • Description: A unique integer value to hold the client of a given appointments ID. • Type: INT • How it's used: Used to link the appointment table to the users table in the application. • How its set: System creates a unique int when each user is created and that will be used to create the clientID of an appointment.
cost	<ul style="list-style-type: none"> • Description: This will be the amount that the client is paying for the appointment. This amount will be determined by the application. • Type: DECIMAL (5, 2) • How it's used: The application will store this amount with the appointment information. The stylist will be able to view this amount. • How its set: The system will generate this based on the time and services.
dateAdded	<ul style="list-style-type: none"> • Description: Keeps track of the times and dates various entries are created. • Type: DATETIME • How it's used: The application will use this to know when a particular entry was added to the database. • How its set: The application will automatically generate the date added in order to track when an entry was created.
dateCanceled	<ul style="list-style-type: none"> • Description: Holds the date of when an appointment was cancelled. • Type: DATETIME • How it's used: Used to keep track of when the appointment was cancelled in the application. • How its set: When a user selects to cancel an appointment the application will automatically generate this value.

dateModified	<ul style="list-style-type: none"> • Description: Holds the time and date of the last change made to particular entry. • Type: DATETIME • How it's used: Used to keep track of when the last modification was made on a particular entry. • How its set: System will generate the DATETIME when an entry is modified in the application.
dateReviewed	<ul style="list-style-type: none"> • Description: Keeps track of the date and time that an appointment is reviewed and approved or denied. • Type: DATETIME • How it's used: The application will use this to know when a particular appointment was reviewed. • How its set: The application will generate this when a stylist review a pending appointment.
day	<ul style="list-style-type: none"> • Description: Stores the name of the day. Monday to Sunday • Type: DATETIME • How it's used: Used to know the different days of the week and their spellings. • How its set: Hardcoded into the application.
description	<ul style="list-style-type: none"> • Description: Holds the different services that will be provided by the stylist for the appointment. • Type: VARCHAR(1024) • How it's used: This will be used by the application to determine which service the client wants to have done. It will help also determine the cost of the particular appointment. • How its set: The user will enter this information when the appointment is created. • Validation: The application will validate the inputs from the user and prompt if there was an incorrect entry.

email	<ul style="list-style-type: none"> • Description: An email address associated with a particular account. • Type: VARCHAR(45) • How it's used: Identifying attribute for an account and how a user will login and be contacted. • How its set: During the creation of an account, an email address will be used and associated with. • Validation: The UI will validate that a set of characters, a (at) sign, another set of characters, a (dot) and finally a CANN domain such as: com, net, org...
endDateTime	<ul style="list-style-type: none"> • Description: The ending date of a particular appointment. • Type: DATETIME • How it's used: Specifies the ending date for a particular appointment. • How its set: When an appointment is created the system will generate an ending date for the appointment.
endTime	<ul style="list-style-type: none"> • Description: Hold the end time of an event. This will be the final date of a particular thing. • Type: TIME • How it's used: Used to determine the end date of a particular event such as time off or a particular schedule. • How its set: Set by the user creating the event. • Validation: The application will ensure that validation is a correct date in the future.

firstName	<ul style="list-style-type: none"> • Description: The specific users first name • Type: VARCHAR(45) • How it's used: Used to refer to the person either on the landing page of the website or in the email header when sending out for contact. It will also be used to identify people to the stylist on his/her schedule. • How its set: Each user will set his/her name during the account creation process. Stylist and Admin accounts firstname is set during the creation of their accounts by the admin user. • Validation: The UI will validate that the name is in fact a valid string.
hairColor	<ul style="list-style-type: none"> • Description: Holds the hair color of an user. • Type: VARCHAR(45) • How it's used: Used to show the hair color of a user on their profile. • How its set: Upon profile creation a user will enter their hair color that the application saves. • Validation: The system will validate the input is a color of hair.
hairLenght	<ul style="list-style-type: none"> • Description: This element will be an optional descriptor to a client's hair length • Type: VARCHAR(45) • How it's used: It is used by the stylist to gauge if the given amount of time for a given hair appointment is appropriate. • How its set: Set when the client chooses to modify and provide further information for his/her profile. • Validation: Any string is valid for this field. The application will filter all escape characters to protect against SQL injections.

hairStyle	<ul style="list-style-type: none"> • Description: The hairstyle of a client. • Type: VARCHAR(45) • How it's used: Used to store the style of hair a client has so the stylist knows what will be needed for the appointment • How its set: The user enters their hairstyle into a field on the profile page
isActive	<ul style="list-style-type: none"> • Description: A bit that holds whether or not a particular user's account is active. If the account is inactive it cannot be logged into. • Type: BIT • How it's used: This bit is used by the application determine whether or not an account is active. • How its set: Admins can set if an account isActive bit is set or not. • Validation: Application generated and therefore will always be valid.
isApproved	<ul style="list-style-type: none"> • Description: A bit that stores whether or not an appointment has been approved by the stylist. • Type: BIT • How it's used: The application will use this to determine whether or not a stylist has approved any given appointment made by a client. • How its set: The application will set this bit based on the stylists review of the appointment.

isCanceled	<ul style="list-style-type: none"> • Description: Indicates if a given appointment is cancelled or not. • Type: BIT • How its used: The client, after creating his or her appointment and getting a notification of approval, can then cancel his or her appointment. This will set a 1 or a 0 if the application is cancelled or not. • How its set: Set when the stylist or client chooses to cancel a given appointment.
lastName	<ul style="list-style-type: none"> • Description: The specific users last name • Type: VARCHAR(45) • How it's used: Used to refer to the person either on the landing page of the website or in the email header when sending out for contact. It will also be used to identify people to the stylist on his/her schedule. • How its set: Each user will set his/her name during the account creation process. Stylist and Admin accounts last name is set during the creation of their accounts by the admin user. • Validation: The UI will validate that the name is in fact a string.
modifiedBy	<ul style="list-style-type: none"> • Description: The last user to modify a user's notes. • Type: INT • How it's used: A foreign key linking to a user in the users table. • How its set: Set by the system when a note is edited.
note	<ul style="list-style-type: none"> • Description: A set of notes left by the user or the stylist. • Type: VARCHAR(1024) • How it's used: to help provide a better environment. • How its set: When the appointment is made by any user.

noteID	<ul style="list-style-type: none"> • Description: A primary key for a user note entry. • Type: INT • How it's used: Used to find the user note. • How its set: Set by the system when the note is created.
password	<ul style="list-style-type: none"> • Description: All users salted hash of their passwords. • Type: VARCHAR(160) • How it's used: To login to the system. • How its set: When the account is created.
passwordResetID	<ul style="list-style-type: none"> • Description: A system generated string of random characters. • Type: VARCHAR(1024) • How it's used: Used when a user attempts to reset their password. • How its set: When a user generates a password reset request, this string is generated by the system.
passwordSalt	<ul style="list-style-type: none"> • Description: The original salt of all of the passwords. • Type: VARCHAR(160) • How it's used: To login to the system. • How its set: When the account is created.
paymentType	<ul style="list-style-type: none"> • Description: This is the type of payment used by a client for a particular appointment. • Type: VARCHAR(45) • How it's used: The application will store this type so the stylists or client can view it in the appointment history. • How its set: This will be set by the stylist when the client makes a payment.

paymentTypeID	<ul style="list-style-type: none"> • Description: The id used as a key for the payment Types table. • Type: INT • How it's used: The application will use this to tie an appointment to the payment for that appointment in the database. • How its set: The application will generate this int when the appointment is created.
phone	<ul style="list-style-type: none"> • Description: The primary phone number stored for a user. This will be the main form of contact used besides email. • Type: VARCHAR(10) • How it's used: The application will store this number with the users profile and will be displayed to any stylist viewing the profile. • How its set: The user will enter this number when they update or create a profile. • Validation: The application will validate the input of the user as a valid phone number.
phone2	<ul style="list-style-type: none"> • Description: Users can enter a secondary phone number that they can be contacted act if they are unreachable at the first one. • Type: VARCHAR(10) • How it's used: The application will store this number with the users profile and will be displayed to any stylist viewing the profile. • How its set: The user will enter this number when they update or create a profile. • Validation: The application will validate the input of the user as a valid phone number.

picture	<ul style="list-style-type: none"> • Description: Picture of hair of a user or a picture of hair done by the stylists. These are displayed on the users profile page. • Type: MEDIUMBLOB • How it's used: Used by the application to show a stylists a picture of a client's hair or clients can example works of a stylists work. • How its set: The user will upload the picture to the system when they update or create their profile. • Validation: The application will validate that the image uploaded by the user.
scheduleID	<ul style="list-style-type: none"> • Description: A reference to the schedule table • Type: INT • How it's used: To associate a stylist with the proper appointment • How its set: When a stylist designates a working window, this element will be populated by the system.
startDateTime	<ul style="list-style-type: none"> • Description: Denotes the start date of an appointment • Type: DATETIME • How it's used: To ensure that the correct blocks of times are closed off if an appointment is made. Also, to keep track of which client is being service by which stylist. • How its set: During the creation of the appointment by the client.
startTime	<ul style="list-style-type: none"> • Description: Denotes the start time of an appointment • Type: TIME • How it's used: To ensure that the correct blocks of times are closed off if an appointment is made. Also, to keep track of which client is being service by which stylist. • How its set: During the creation of the appointment by the client.

stylistID	<ul style="list-style-type: none"> • Description: An ID to associate a user whose member type denotes stylist to a stylist table • Type: INT • How it's used: To keep track of the stylists • How its set: When a stylist is created, the system will assign an integer to each stylist.
stylistsXAppointmentTypeID	<ul style="list-style-type: none"> • Description: Value used to reference the appointment with the appointment type. • Type: INT • How it's used: Used by the application to link the appointments, stylistsXAppointmentTypes and appointmentTypes. • How its set: The system generates this int when an appointment is created.
userID	<ul style="list-style-type: none"> • Description: A unique ID given to a user whose member level denotes a regular user (client). • Type: INT • How it's used: To associate users to their proper table with minimal overhead. • How its set: During the creation of the account, they are assigned by the system.
userType	<ul style="list-style-type: none"> • Description: An identifier to user types. eg: Admin, Stylist or Client • Type: INT • How it's used: Used to determine a user's access levels to different feature of the system. • How its set: Hardcoded into the database.

userId	<ul style="list-style-type: none">• Description: The primary key for the users table.• Type: INT• How it's used: Used to look up the user related to a post.• How its set: Hardcoded into the database.
--------	--

Data Tables

A list of tables in our database including all the data elements used by each tables as well as their size and estimated growth rates.

Table	Meta
appointments	<ul style="list-style-type: none">• Description: The table used to store appointments between clients and stylists.• How it is set: When a new appointment is created a new row is added to the• How it is used: Used to look up information related to specific appointments.• Number of "records": 0• Expected growth in number of "records": 10 per day per stylist• "Record" size in bytes: 113• Composition<ul style="list-style-type: none">○ clientID - 4 bytes○ stylistsXAppointmentTypeID - 4 bytes○ modifiedBy - 4 bytes○ appointmentID - 4 bytes○ paymentType - 45 bytes○ startDateTime - 8 bytes○ endDateTime - 8 bytes○ dateAdded - 8 bytes○ dateCanceled - 8 bytes○ dateModified - 8 bytes○ dateReviewed - 8 bytes○ ammountPaid - 4 bytes○ cost - 3 bytes○ isApproved - 1/8 bytes○ isCanceled - 1/8 bytes

appointmentTypes	<ul style="list-style-type: none"> • Description: Holds the services that are going to be done at a specific appointment. • How it is set: Created when a user creates an appointment. • How it is used: Used by the application to store the different services that will be done at an appointment. • Number of "records": 10 • Expected growth in number of "records": 10+ year • "Record" size in bytes: 1040 bytes • Composition <ul style="list-style-type: none"> ○ appointmentTypeID - 4 bytes ○ description - 1024 bytes ○ dateAdded - 8 bytes ○ motifiedBy - 4 bytes
days	<ul style="list-style-type: none"> • Description: A list of static entries for the different days of the week. • How it is set: All entries in this table are set up when the system is created. • How it is used: Used to set up each stylist's availability for each day of the week. • Number of "records": 7 • Expected growth in number of "records": 0. It will not grow. • "Record" size in bytes: 45 bytes • Composition <ul style="list-style-type: none"> ○ day - 45 bytes

paymentTypes	<ul style="list-style-type: none"> • Description: Stores the type of payment in which the appointment was paid for. • How it is set: The application will generate this when a user creates and appointment. • How it is used: Used by the application to determine the payment type used for a given appointment. • Number of "records": 4 • Expected growth in number of "records": 4 • "Record" size in bytes: 94 bytes • Composition <ul style="list-style-type: none"> ○ paymentTypeID - 4 bytes ○ paymentTypeType - 45 bytes ○ paymentSubType - 45 bytes
stylistsDayOffs	<ul style="list-style-type: none"> • Description: Indicates a day where a stylist has no availability. • How it is set: The stylist would indicate that they wish to take a certain day off. • How it is used: A day off will set a stylists availability to none for that day without affecting the stylist's schedule. • Number of "records": 0 • Expected growth in number of "records": 2 per month • "Record" size in bytes: 22 • Composition: <ul style="list-style-type: none"> ○ scheduleID - 4 bytes ○ stylistID - 4 bytes ○ startTime - 3 bytes ○ endTime - 3 bytes ○ dateAdded - 8 bytes

stylistsSchedules	<ul style="list-style-type: none"> • Description: Contain that times for each day that the stylists plans to work. • How it is set: The stylist enters the hours they are going to work each day. • How it is used: The application will use this to determine the availability of each stylist. • Number of "records": 3 • Expected growth in number of "records": 1+ for each new stylist • "Record" size in bytes: 59 • Composition <ul style="list-style-type: none"> ○ scheduleID - 4 bytes ○ stylistID - 4 bytes ○ day - 45 bytes ○ startTime - 3 bytes ○ endTime - 3 bytes
stylistsXAppointmentTypes	<ul style="list-style-type: none"> • Description: A cross table that links stylists with appointment types. • How it is set: When a stylist adds types of appointments they can service a new row is created. • How it is used: Used to link a stylist with an appointment type • Number of "records": 0 • Expected growth in number of "records": ~10 per stylist • "Record" size in bytes: 27 1/8 • Composition <ul style="list-style-type: none"> ○ stylistXAppointmentTypeID - 4 bytes ○ stylistID - 4 bytes ○ appointmentTypeID - 4 bytes ○ isActive - 1/8 bytes ○ dateAdded - 8 bytes ○ modifiedBy - 4 bytes ○ cost - 3 bytes

userNotes	<ul style="list-style-type: none"> • Description: Each entry represents the notes entered into a particular field. • How it is set: The user will enter the desired notes for a particular field. • How it is used: The application will allow the user to view notes on an appointment or profile. • Number of "records": 0 • Expected growth in number of "records": 1+ per new client • "Record" size in bytes: 1048 • Composition <ul style="list-style-type: none"> ○ modifiedBy - 4 bytes ○ noteID - 4 bytes ○ userID - 4 bytes ○ appointmentID - 4 bytes ○ note - 1024 bytes ○ dateAdded - 8 bytes
userTypes	<ul style="list-style-type: none"> • Description: A static table in the database containing an entry for each of the different types of users (Admin, Stylist, Client) • How it is set: The entries in the table will be added when the system is created. • How it is used: Used to grant permissions for each user. • Number of "records": 4 • Expected growth in number of "records": 0 • "Record" size in bytes: 49 • Composition <ul style="list-style-type: none"> ○ userTypeID - 4 bytes ○ UserType - 45 bytes

users	<ul style="list-style-type: none"> • Description: A table with a list of users for the system. • How it is set: When a new user is created a new row is added. • How it is used: Used to look up and interact with users. • Number of "records": 9 • Expected growth in number of "records": 1+ per day. • "Record" size in bytes: 2177 • Composition <ul style="list-style-type: none"> ○ userID - 4 bytes ○ userTypeID - 4 bytes ○ modifiedBy - 4 bytes ○ lastName - 45 bytes ○ firstName - 45 bytes ○ email - 45 bytes ○ isActive - 1/8 bytes ○ dateAdded - 8 bytes ○ dateModified - 8 bytes ○ password - 160 bytes ○ passwordSalt - 160 bytes ○ passwordResetID - 1024 bytes ○ picture - $L + 3$ bytes, where $L < 2^{24}$ ○ phone - 10 bytes ○ phone2 - 10 bytes ○ hairStyle - 45 bytes ○ hairColor - 45 bytes ○ hairLength - 45 bytes
-------	--