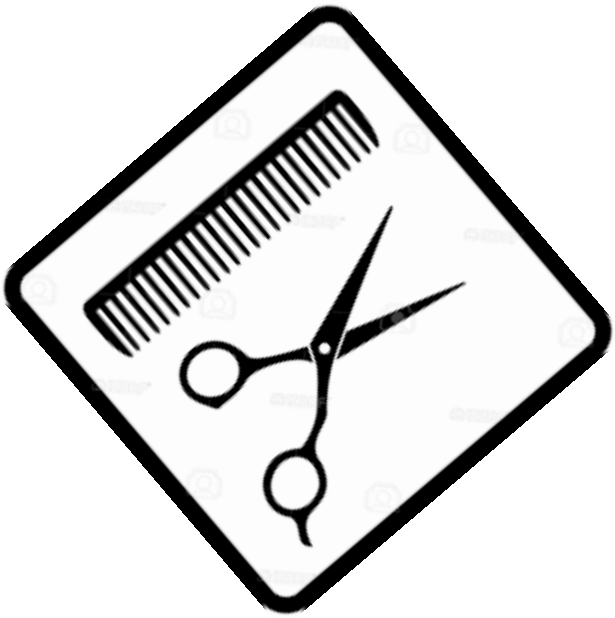
SRS REPORT



GROUP NO: 9

MEMBERS: SAHIL SHARMA

MADHAV MALPANI

DUSHYANT KAUSHIK

UJJVAL RANA

DATE: 6/9/17

TABLE OF CONTENTS

1. INTRODUCTION
    1.1 Purpose (3)
    1.2 Scope Of Project (3)
    1.3 Tools to be used (3)
    1.4 Technologies to be used (3)
    1.5 References (4)
    1.6 Overview (4)

2. OVERALL DESCRIPTION
    2.1 Product Description (5)
    2.2 Product Features (6)
    2.3 Operating Environment (6)
    2.4 Design and Implementation Constraints (6)
    2.5 Assumption and Dependencies (7)

3. SYSTEM FEATURES (8)

4. EXTERNAL INTERFACE REQUIREMENTS
    4.1 User Interfaces (9)
    4.2 Hardware Interfaces (9)
    4.3 Communication Interfaces (9)
    4.4 Software Interfaces (10)

5. OTHER NON-FUNCTIONAL REQUIREMENTS
    5.1 Performance Requirements (11)
    5.2 Safety Requirements (11)
    5.3 Security Requirements (11)
    5.4 Software Quality Attributes (11)

INTRODUCTION

1.1 Purpose

The purpose of this document is to provide the software requirement specification report for our software project: "Belleza scheduling web app". The requirements will vary a lot depending on the kind of architecture we are using to build this platform while introducing new features and additional functionalities. We've already been through our predevelopment phase and this is supposed to be our initiation of the development phase.

1.2 Scope Of Project

- There are two basic users – Belleza Owner and Students
- Both users will have different programs
- A robust appointment scheduling system for students
- Belleza owner will be able to see beforehand the number of appointments for the days and be prepared accordingly.
- Students will be able to view real time status (seat available or not) for different tasks at Belleza.

1.3 Tools To Be Used

- Flask: The Web framework for perfectionists (with deadlines). Flask makes it easier to build better Web apps more quickly and with less code. Flask is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It lets you build high-performing, elegant Web applications quickly. Flask focuses on automating as much as possible and adhering to the DRY (Don't Repeat Yourself) principle.

Flask is best for:-

- Developers who aim to include all the batteries a web application will need.
- Developers who wants to use ORM, templating, routing, authentication, database administration and bootstrapping on the go.

1.4 Technologies To Be Used

Front-end web development: It is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

INTRODUCTION

- We'll use python 3.5 to implement this feature. Another framework which we can use for the same is Flask, based on python which provides native support for streaming responses through the use of generator functions. A generator is a special function that can be interrupted and resumed. With this kind of Streaming it is possible to generate a large data table, without having to assemble the entire table in memory.

- We'll use HTML 5.1's new features in our web designing phase, using new features such as geo location, geo tagging, client storage facilitation and better JavaScript integration with Bootstrap for styling the web pages and making web app for more elegant.

- As we'll try to launch this application on a small scale, in this case our college, having our own distinct 'SNU' Gmail ID's might let the integration of Google chat with rich Ajax interface. One alternative to this is to create an Instant Messaging app using Google Cloud Messaging (GCM) using PyDev is a plugin that enables Eclipse to be used as a Python IDE (supporting also Jython and IronPython).

- The web based interface we'll design for our project will involve rich background, good social networking features. We're going to look and observe popular discussion forums, video streaming and social networking sites to implement a better system when we write our own code.

1.5 References

- IEEE.*IEEE std 830-1998 IEEE Recommended Practice for Software Requirements*
- https://wiki.python.org/moin/WebFrameworks
- https://www.quora.com/What-is-the-best-Python-web-app-framework-And-why
- https://www.drupal.org/project/googlechat
- https://en.wikipedia.org/wiki/Motion_JPEG
- https://developers.google.com/youtube/v3/live/code_samples/python

1.6 Overview

- Our Plan: Live Status of Belleza Saloon
  Live Booking and status check
  Cancellation of booking

OVERALL DESCRIPTION

2.1 Product Description

The Web based appointment system is going to be a browser based application which would enhance the saloon experience and make it more hassle free. Features like real time status, management of people, checking the costs of different services which are present in saloon which will be improvised and improved upon to give a comfortable user experience. The software would improve user experience for both: Belleza owner and student.

Product Functions:

User	Saloon	Status	Reservation	Cancel Appointment
User ID	Service ID	Live status	Reservation ID	Reservation ID
User Name	Service name	Live booking	Service ID	Privileged
User Preferences	Estimated Time		User ID	
Password	Cost		Date Time	

- The two major elements of the product are Belleza Owner and Student.
- The student can see live status. User gets notified anytime a chair gets empty if he has notifications enabled. The Belleza owner can himself clear the chairs according to if the person came or cancelled the appointment
- Users will be able to book appointment beforehand and thus will have a hassle free experience.
- Diff. service reviews is also a part of the application.
- The Belleza owner can himself clear the status of different chairs according to him. This will be reflected in real time to other users
- Students on the days of massive rush can pre book and even see real time status so they don't have to go there and wait for infinite time.

2.2 Product Features

- Real time status check
- Pre book appointment upto next 2 days
- People who booked the list along with what service for the Belleza owner
- Check cost of different services
- A different review page for first hand reviews of diff. services
- 100% web based for clients

2.3 Operating Environment

Operating environment for Web based Booking system is as listed below:

- centralised database system design
- client/server system for real time data transfer
- operating system : compatible with any web browsers after IE8
- database: sql+ database
- platforms: Multi-platform support

2.4 Design And Implementation Constraints

- The free servers are not feasible enough to support real time data for all the day.
- A server must be running all the time for booking and checking of appointments.
- Flask must be learnt to implement the servers and application
- UI should be smooth as the it is integral part of the application

2.5 Assumption And Dependencies

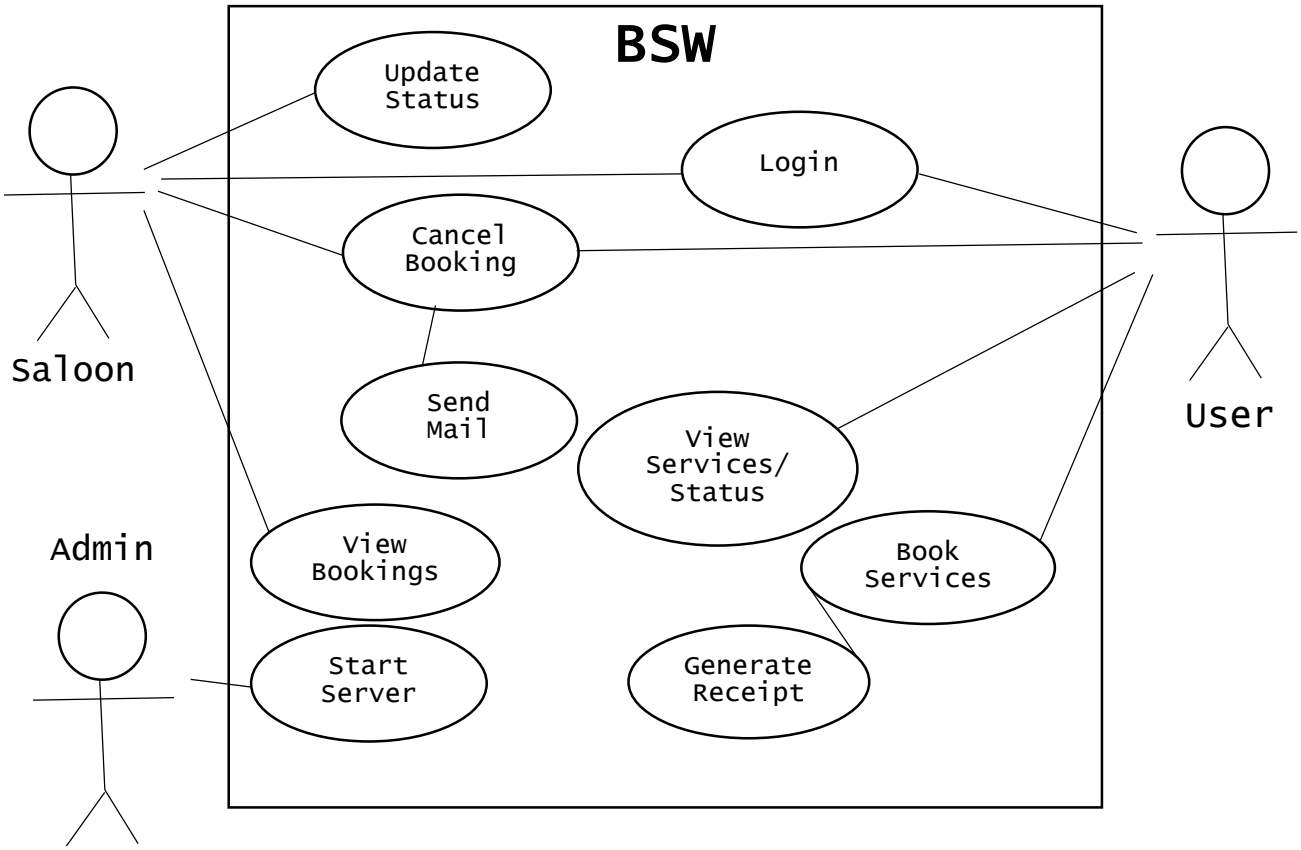
- The feature of broadcasting to so many users at one time involves getting server malfunctions because of bad tunnelling, runtime architectural problems and cross platform integration errors
- We need a good internet connection to facilitate our services to the users
- Some browsers extensions might not work as they're supposed to be due to lack of support
- Server maintenance on regular intervals

SYSTEM FEATURES

- Description and Priority : The system allows user to book an appointment for the saloon. The user can choose what all services he or she wants in advance upto 24 hours from the webapp. The person operating the saloon will receive all the bookings and can be prepared in advance. Saves a lots of time.

- Stimulus/Response Sequences :The operator of salon can view the bookings in advance. If he is not available at that time, he can cancel the booking and the user will get a mail regarding cancelation of his or her appointment. If the user does not come to the salon during the appointment time, after 15 minutes of the starting time of appointment the appointment will get cancelled automatically.

Functional requirements:



Use Case Name	Start server
Trigger	Admin boots up the server to accept requests
Precondition	The database is ready and no garbage/testing values are visible
Basic Path	1. Load the webapp to host device 2. Start up the server 3. Accept requests
Alternative Paths	2.1. Start debug mode to configure errors and bugs
Post condition	The webapp is running and accepting request
Exception Paths	Server may stop working due to inactivity and poor maintenance
Other	Database setup and testing is completed Actor involved is Admin

Use Case Name	Login
Trigger	User authenticate to access webapp
Precondition	User is navigated to webapp
Basic Path	1. User enters credentials and password 2. Credentials are matched to their roles 3. User is authenticated to use the app
Alternative Paths	1.1. User is a guest user 1.2. [redirect to view status]
Post condition	[redirect to view status]
Exception Paths	1.1.1. User doesn't have valid credentials 1.1.2. User creates an account 1.1.3. [redirect to login] 1.2.1. User forgot her password 1.2.2. user requests for change of password/credentials 2.1. credentials do not match 2.2. [redirect login]
Other	Actors involved are User and Saloon

Use Case Name	View Status/Services
Trigger	Live status of the saloon can be accessed
Precondition	User is logged in.
Basic Path	1. User can see current status of saloon 2. User select date-time to view status
Alternative Paths	1.1. User can see available services 2.1 User can log-out
Post condition	
Exception Paths	2.1 User selects out-of-range date 2.2 A warning is prompted
Other	Logged in user can be a guest user Actor is user

Use Case Name	Book service(s)
Trigger	User clicked book link
Precondition	User is logged in
Basic Path	1. User selects date-time as his slot 2. User selects services 3. Proceeds to confirm booking
Alternative Paths	1.1. User selects book now 1.2. Proceeds to confirm booking
Post condition	Generate receipt
Exception Paths	1.1. Date-time is not a valid date 1.2. A warning is prompted 2.1 User presses back/cancel button 2.2 Return to view status
Other	Actor is user Actor need to be logged in No guest user allowed

Use Case Name	Generate receipt
Trigger	User clicked confirm booking
Precondition	Booking form is filled correctly
Basic Path	1. Cost is calculated 2. Estimated time is calculated 3. An acknowledgement receipt is generated 4. Terms and conditions for validity of bookings are shown
Alternative Paths	
Post condition	Database contains new booking record Status for booked slot changed A reminder mail is sent to user
Exception Paths	1.1. No free slots are available 1.2. A warning is prompted to user 1.3. Redirect to book services
Other	

Use Case Name	Cancel Booking
Trigger	User clicked cancel booking
Precondition	User has visited cancel booking
Basic Path	1. User selects a valid receipt 2. Clicks on confirm cancel 3. Acknowledgement is received
Alternative Paths	3.1. User is saloon 3.2. Can cancel multiple(all) reservations
Post condition	Booking record is removed from the database Status for booked slot is changed
Exception Paths	1.1.1. Receipt is of expired time slot

	1.1.2. A warning is prompted 1.2.1. User is not authorized user to cancel booking 1.2.2. A warning is raised 1.2.3. Redirect to view status
Other	Actor must be receipt owner Actor can be Saloon

Use Case Name	Send mail
Trigger	User has cancelled an appointment
Precondition	Appointment cancellation was successful
Basic Path	1. The owner of receipt is found 2. A pre-drafted mail is sent to the found user 3. A successful message is displayed
Alternative Paths	3.1. In case of multiple cancellation a day-off option is asked 3.2. If option is set to true, no new reservation for the slot is allowed
Post condition	Booking record(s) is removed from the database Status for booked slot changed
Exception Paths	1.3. The owner of the receipt is not found
Other	Actor must be Saloon

Use Case Name	View booking
Trigger	Saloon has logged in
Precondition	User login was successful
Basic Path	1. A list of upcoming appointments are visible 2. Corresponding status is changed
Alternative Paths	1.1. No upcoming appointments are there 1.2. Current status of 'Now Open' shown
Post condition	Current status is modified
Exception Paths	
Other	Actor must be Saloon

Use Case Name	Update Status
Trigger	User has clicked an appointment
Precondition	Actor logged in
Basic Path	1. A new customer has arrived 2. Change the status to busy
Alternative Paths	1. A customer has left 2. Change the status to free
Post condition	Current status is modified
Exception Paths	1.1. The shop is closed 1.2. Change status to busy for all day
Other	Actor must be Saloon

EXTERNAL INTERFACE REQUIREMENTS

4.1 User Interfaces

Front end softwares: HTML, CSS, Javascript

Back end softwares: Python, Flask, SQL+

4.2 Hardware Interfaces

Minimum requirement :

- Web Browser
- Ram (128 MB)
- Disk Space (100 MB)
- All Intel or AMD ~ 800 MHz
- Internet Speed ~ 0.5 Mbps (preferred)

4.3 Communication Interfaces

- This project supports all types of web browsers
- We are using rich interface build to sustain many no of simultaneous requests.
- Using geo location and Google cloud messaging services we are making use of cloud based storage too

4.4 Software Interfaces

Softwares used	Description
Operating system	Any operating system will do.
Database	To save the entries and real time requests we will be using sqlite along with sql on the server.
Flask (python based web framework)	To implement the project we have chosen Python language for its more interactive support.
HTML,CSS(bootstrap),Javascript	For designing and furnishing our website.

OTHER NON-FUNCTIONAL REQUIREMENTS

5.1 Performance Requirements

The basic objective is to be reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a Database is not properly designed it can gives rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table.

5.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.4 Software Quality Attributes

- Availability: The real time status should be readily available and there should be no lag while booking an appointment.
- Correctness: There should be no clashes as such for the same time slot.
- Maintainability: The Belleza saloon owner should be clearing all the appointments done till time so as to clear the slot for the next person.
- Usability: The real-time status of the services available, pre booking for services and the costs corresponding to them should be enough to justify the usability of this web app.