**SHIV NADAR UNIVERSITY**

CSD301 - SOFTWARE ENGINEERING

# Final Project Report

## Belleza Scheduling Web Application

**Submitted By: Group 9**

**Date: 17th November 2017**

**Guided By: Prof. Snigdha Biswas**

# DEVELOPMENT TEAM

Madhav Malpani

Sahil Sharma

Ujjval Rana

Dushyant Kaushik

# TABLE OF CONTENTS

# Project Proposal

## PROJECT TITLE: BELLEZA SCHEDULING WEBAPP.

## PROJECT IDEA:

A 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 students.

## OBJECTIVES:

- Easier and faster way to book appointments.
- Minimise the waiting time.
- Creating a user- friendly way to to do.

## THE MAIN FUNCTIONALITIES OF THE SYSTEM SHALL BE AS FOLLOWS:

- Our system is basically a webapp which can run on any platform.
- The Belleza owner keeps the server side updated if he currently has any booking.
- He/she may also fill in any future appointments made by a student(in person), so it also acts a database for the owner.
- When a student logs in, he/she can view the live status of the saloon and get to know if any seats are available, and book accordingly.
- After logging in, the user may book any time slot in advance, the database of the saloon owner gets updated as well. This ensures no one else can book his slot at that particular time.

- The students or the owner may also cancel the appointments due to any unforseen circumstances and the a notification is sent to the other(student or saloon) immediately, notifying them of the cancellation.

## ESTIMATED PROJECT SCHEDULE:

*Estimated Start date*: 12th August 2017

*Estimated End Date*: 20th November 2017

# SRS REPORT

# TABLE OF CONTENTS

## 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.

- 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 up to 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:

- centralized 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

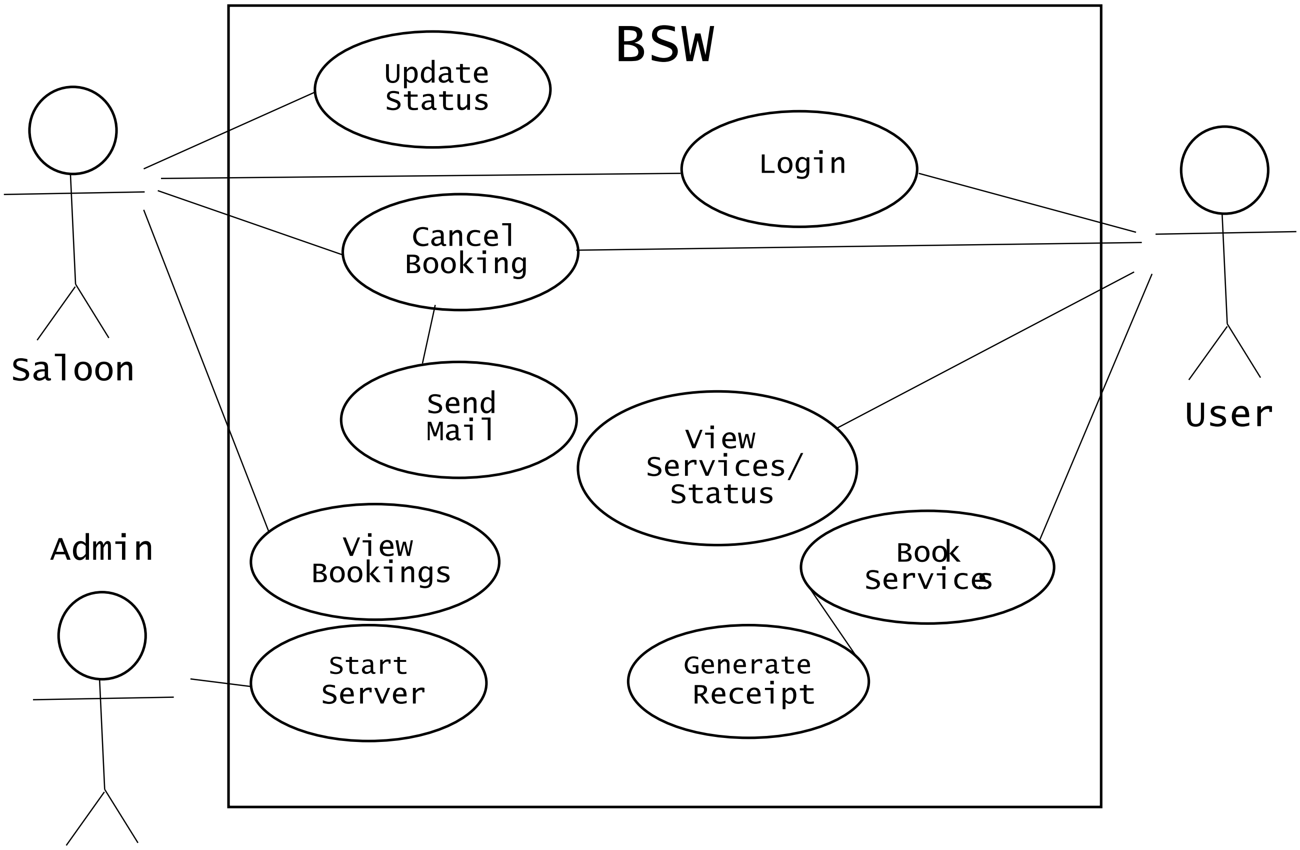
- The feature of broadcasting to so many users at one time involves getting server malfunctions because of bad tunneling, runtime architectural problems and cross platform integration errors.
- We need a good internet connection to facilitate our services to the users.
- Some browsers extensions might 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<br>2. Start up the server<br>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<br>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<br>2. Credentials are matched to their roles<br>3. User is authenticated to use the app |
| **Alternative Paths** | 1.1. User is a guest user<br>1.2. [redirect to view status] |
| **Post condition** | [redirect to view status] |

| | |
|---|---|
| **Exception Paths** | 1.1.1. User doesn’t have valid credentials<br>1.1.2. User creates an account<br>1.1.3. [redirect to login]<br>1.2.1. User forgot her password<br>1.2.2. user requests for change of password/credentials<br>2.1. credentials do not match<br>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.<br>User select date-time to view status |
| **Alternative Paths** | 1.1. User can see available services<br>2.1 User can log-out |
| **Post condition** | |
| **Exception Paths** | 2.1 User selects out-of-range date<br>2.2 A warning is prompted |
| **Other** | Logged in user can be a guest user<br>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<br>2. User selects services<br>3. Proceeds to confirm booking<br>4. |
| **Alternative Paths** | 1.1. User selects book now<br>1.2. Proceeds to confirm booking |
| **Post condition** | Generate receipt |
| **Exception Paths** | 1.1. Date-time is not a valid date<br>1.2. A warning is prompted<br>2.1 User presses back/cancel button<br>2.2 Return to view status |
| **Other** | Actor is user<br>Actor need to be logged in<br>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<br>2. Estimated time is calculated<br>3. An acknowledgement receipt is generated<br>4. Terms and conditions for validity of bookings are shown |
| **Alternative Paths** | |
| **Post condition** | Database contains new booking record<br>Status for booked slot changed<br>A reminder mail is sent to user |
| **Exception Paths** | 1.1. No free slots are available |

| | |
|---|---|
| | 1.2. A warning is prompted to user<br>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<br>2. Clicks on confirm cancel<br>3. Acknowledgement is received |
| **Alternative Paths** | 3.1. User is saloon<br>3.2. Can cancel multiple(all) reservations |
| **Post condition** | Booking record is removed from the database<br>Status for booked slot is changed |
| **Exception Paths** | 1.1.1. Receipt is of expired time slot |
| | 1.1.2. A warning is prompted<br>1.2.1. User is not authorized user to cancel booking<br>1.2.2. A warning is raised<br>1.2.3. Redirect to view status |
| **Other** | Actor must be receipt owner<br>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<br>2. A pre-drafted mail is sent to the found user<br>3. A successful message is displayed |
| **Alternative Paths** | 3.1. In case of multiple cancellation a day-off option is asked<br>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<br>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<br>2. Corresponding status is changed |
| **Alternative Paths** | 1.1. No upcoming appointments are there<br>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<br>2. Change the status to busy |

| | |
|---|---|
| **Alternative Paths** | 1. A customer has left<br>2. Change the status to free |
| **Post condition** | Current status is modified |
| **Exception Paths** | 1.1. The shop is closed<br>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.

# Software Project Management Plan

# CONTENTS

# 1. INTRODUCTION

## 1.1 PROJECT OVERVIEW:

BSW is a multi-platform web application that focuses on aiding the saloon by providing them with a scheduling system. This system will serve as a useful interface between the customers and the saloon. BSW is one such application, which is simple to use and has an interactive user interface.

Being a stand-alone application, this software will be centrally governed. All the clients will have access to the system but shall be managed by one admin only.

Clients can cancel the appointment if he/she wishes to. Overall this project is about helping the clients to easily get their appointments without them having to wait in the saloon. Also, it helps the saloon to get a better idea of number of clients they might have the following day.

## 1.2 PROJECT DELIVERABLES:

The project aims to deliver a working application, which works on multi- platform (stand-alone and web). It has a verified login and guarantees security.

With stand-alone application, only admin gets to edit/delete the current status information of the saloon. However, a user can cancel their appointments as well.

# 2. PROJECT ORGANIZATION

## 2.1. SOFTWARE PROCESS MODEL:

The BSW project will follow an incremental and an iterative development model for its deliverables.

The development will be done in several phases and each phase will represent a complete development cycle, with certain functionality of the system delivered at the end of each phase. The phased approach to delivery provides flexibility in what the team will deliver, gives an opportunity to reassess the effort for each phase and allows both the team and the client to change any of the phase's content.

## 2.2.2. ROLES AND RESPONSIBILITIES:

Projects of different sizes have different needs for how the people are organized. In a small project, little organization structure is needed. However, for large projects, there are more and more people involved, and it is important that people understand what they are expected to do, and what role people are expected to fill.

This section identifies some of the project roles that may be required for the project.

### Software Analyst:

- By definition, a software analyst is a person who studies the software application domain and prepares the software requirements and specification (SRS) document. The analyst is expected to do the analysis, validate it with the client and felicitate the requirements, and convert them into an architecture and design that will become the blueprint for the solution.
- He should possess communication and relationship skills, as well as skills pertaining to word processing apps, spreadsheets.

BACK END DEVELOPER:

- A back-end web developer is responsible for server-side web application logic and integration of the work front-end web developers do. Back-end developers usually write web services and APIs used by front-end developers and mobile application developers.

FRONT END DEVELOPER:

- Front end developers use HTML, CSS, and JavaScript to code the website and web app designs created by web designers. The code they write runs inside the user's browser (as opposed to a backend developer, whose code runs on the web server).

TESTING TEAM:

- The testing team tests the software on the basis of the test cases. The role of the testing team is to develop test cases and prioritize testing activities, execute all the test case and report defects, define severity and priority for each defect and carry out regression testing every time when changes are made to the code to fix defects.

## 2.3. Tools and Techniques:

Tools:

- Programming Languages: HTML, CSS, JAVASCRIPT, PYTHON, SQL.
- Database: SQLite and SQL.
- Design: Object-oriented analysis and design (OOAD)
- Additional: Flask framework hosted on HEROKU Server

Techniques:

- Software Documentation: Documentation such as Software Project Management Plan (SPMP), Software Requirement Specification (SRS), detailed Software Design Document (SDD) and Software Test Document (STD).

# 3. Project Management Plan

## 3.1 Work Breakdown Structure:

| | |
|---|---|
| **Sahil:** | Design, Front-End |
| **Dushyant:** | Back-end,testing |
| **Madhav:** | Back-end,project analyst |
| **Ujjval:** | Testing,Documentation |

## 3.2 Assumptions, Dependencies and Constraints:

### Assumptions:

- Stable internet connection.
- The project team will deliver the work as scheduled.
- The project team will have access to needed systems.
- The required resources will be available throughout the project. Dependencies:

- The information stored in the database can be edited or deleted as and when required.
- Real-Time applicability needs the database to be synchronized.
- Documentation and Testing can happen only along or after the coding.

### Constraints:

- Schedule constraint: To deliver the milestones within the deadline.

- Skills: The team members may not have the skills and capacity to deliver the milestones.
- Resource constraint: All the team members are not available all the time. Conflicts are bound to happen with the individual schedule and project's schedule.
- Database design: The database should be designed to cater to the needs of the software.

## 3.3 TIMELINE

| Phases | Planned Start Date | Planned End Date |
|---|---|---|
| Project Planning | 12/08/2017 | 9/09/2017 |
| Design | 9/09/2017 | 25/09/2016 |
| Coding | 27/09/2017 | 20/10/2017 |
| Unit Testing and Review | 20/10/2017 | 10/11/2017 |
| Documentation | 15/11/2017 | 20/11/2017 |

# 4. Risk Management Plan

## Risk Identification:

| Risk Type | Possible Risks |
|---|---|
| Team | • Conflicts between team members<br>• Members with negative attitude towards project<br>• Lack of cooperation from any member or lack of adequate involvement<br>• Inexperienced team members<br>• Lack of knowledge required for the software development |
| Users | • Failure to manage end user expectations<br>• Admin not satisfied with the robustness of the software |
| Requirements | • Changes to requirements that require major design rework proposed<br>• Customer may fail to understand the impact of requirement change<br>• Misunderstandings of the requirements<br>• Continuous change of requirements |

| | |
|---|---|
| Technology | • Insufficient procedures to ensure security, integrity and availability of the database<br>• Inadequate security features being built into the system<br>• System failure and data loss |
| Tools | □ Software tools cannot work together in an integrated way |
| | □ Possible failure reports not generated on time or correctly. |
| Estimation | □ The time required to develop the software is estimated<br>□ Artificial Deadlines<br>□ Lag on progress due to unavoidable reasons like vacation/exam/member being absent |

## Risk Planning:

| Risk | Strategy |
|---|---|
| Staff Illness | Reorganize team so that there is more overlap of work and people therefore understand each other's jobs. |
| Requirements changes | Derive traceability information to access requirements change impact; |
| System failure | The development team should cease work on that system until the environment is<br>made stable again, or should move to a system that is stable and continue working there. |
| Data Loss | Create regular back - ups of the files and data used in the project. |
| Customer satisfaction | Give Admin sufficient control over the software. |
| Technology Does Not Meet Specifications | The customer should be immediately notified; a meeting should be held between the development team and the customer to discuss at length this issue. |
| Lack of Development Experience | Experienced team members to help the inexperienced team members. |
| Ineffective communication | Hold regular and interactive project meetings. |

## RISK MONITORING:

| Risk Type | Potential Indicators |
|---|---|
| User | Multiple error messages; user/customer complaints |
| Requirements | Many requirements change; confusion w.r.t. requirements in<br>the team |
| Technology | Many reported technology problems; no back-up of files and data created; discrepancies in database; web browser incompatibility |
| Tools | Reluctance by team members to use tools; no error messages generated for exceptions |
| Estimation | Failure to meet agreed schedule; failure to clear<br>reported defects; no progress with the development of<br>software |

# Test Plan Document

# INDEX

# INTRODUCTION

This document is a high-level overview defining our testing strategy for the application. Its objective is to communicate project-wide quality standards and procedures. It portrays a snapshot of the project as of the end of the planning phase. This document will address the different standards that will apply to the unit, integration and system testing of the specified application.

## 1.1 REFERENCE

- SRS
- Software Requirements provided by the client.

## 1.2 FEATURES TO BE TESTED:

- GUI
- Login Test
- Sign Up Test
- Booking Test
- History and Tracking
- Connectivity
- Functionality Test
- Database Test

# APPROACH

1 Functionality Testing

2 Usability Testing

3 Interface Testing

4 Compatibility Testing

5 Performance Testing

6 Security Testing

7 White Box Testing

8 Unit Testing

9 Black Box Testing

# DETAILED EXPLANATION OF APPROACH:

## 2.1 FUNCTIONALITY TESTING-

- **This test is for** – all the links in application pages, database connection, forms used in the application pages for submitting or getting information from user, Cookie testing.

- **Check all the links** - Testing for all outgoing links from all pages under specific domain and also testing the internal links. Testing links jumping on same pages and to check if there are any orphan pages and also checking the broken links.

- **Test forms in all pages**- Forms are the integral part of any web site. Forms are used to get information from users and to keep interaction with them. All the forms of the application were checked during this phase. The default values fields were also checked during this phase of checking. Wrong inputs for the fields were also checked.

- **Database Testing** – The consistency of the data was checked in this phase as it is the most important part of any application. All the functions of the database were tested such as edit, delete, modifying of the database and other functionalities were tested.

## 2.2 USABILITY TESTING-

- **Test for navigation-** In this phase navigation of the Application pages was tested such as buttons, boxes or how user using the links on the pages to surf different pages. Main menu that is provided on the home page was tested to be consistent.

- **Content checking-** Format and quality of the content was checked in this phase. Format of the content was kept in mind such that use of the dark colors and was not used in the Application. The content and the images were checked and placed properly with proper sizes.

- **Other user information for user help-** The tree view of the navigation bar was checked for better usability. Working of all the links were checked in this phase.

## 2.3 INTERFACE CHECKING-

In this phase the interactions between the servers like application server and database server was tested by the testing team. Errors were handled properly.

## 2.4 COMPATIBILITY TESTING-

Compatibility of the Application is an important aspect of the website and the following compatibilities were tested:

**OS compatibility**- Since it is an application designed for Web, it is compatible with all browsers.

## 2.5 PERFORMANCE TESTING-

Application was tested for heavy load and stress which are described below: The Application was checked for large input data and simultaneous connection to database and for heavy load on each page.

## 2.6 SECURITY TESTING-

Several steps were taken to test the security of the website like some invalid inputs were fed to the fields like login and password and hence the security was checked.

## 2.7 White Box Testing-

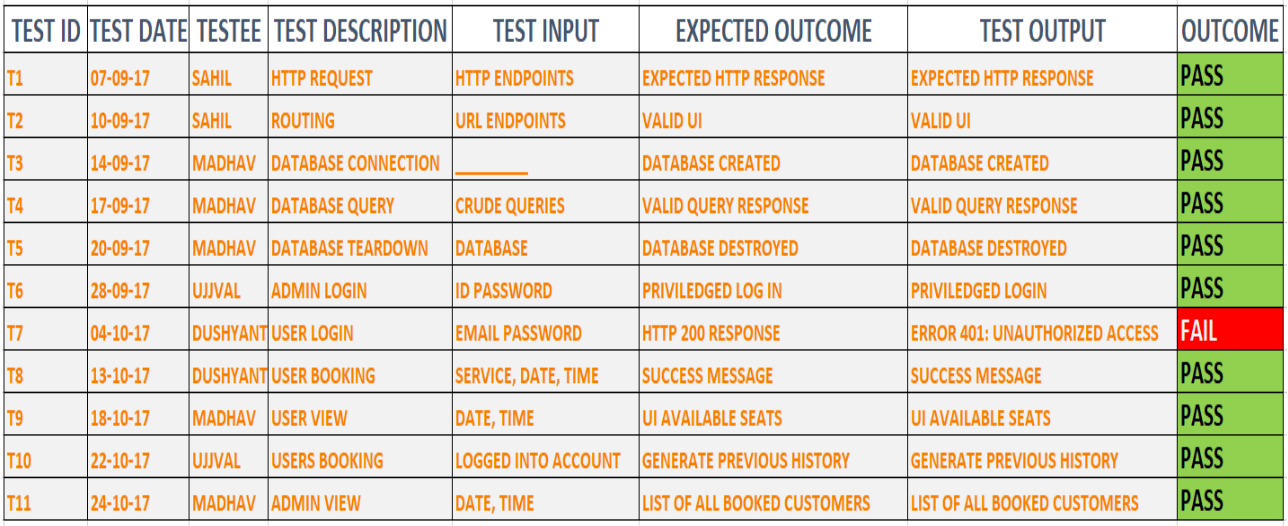
This testing is based on the detailed investigation of internal logic and the structure of the code. This white box testing method was applied to the application and internal working of the code was tested by the tester. All the errors were then rectified.

## 2.8 Black Box Testing-

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. This type of testing was also applied to the application with help of our friends.

## 2.9 Unit Testing-

In this phase of testing individual testing of units of code was done, different modules of the codes were tested separately. Unit testing is beneficial because it makes it easy to maintain the code.



| TEST ID | TEST DATE | TESTEE | TEST DESCRIPTION | TEST INPUT | EXPECTED OUTCOME | TEST OUTPUT | OUTCOME |
|---|---|---|---|---|---|---|---|
| T1 | 07-09-17 | SAHIL | HTTP REQUEST | HTTP ENDPOINTS | EXPECTED HTTP RESPONSE | EXPECTED HTTP RESPONSE | PASS |
| T2 | 10-09-17 | SAHIL | ROUTING | URL ENDPOINTS | VALID UI | VALID UI | PASS |
| T3 | 14-09-17 | MADHAV | DATABASE CONNECTION | ________ | DATABASE CREATED | DATABASE CREATED | PASS |
| T4 | 17-09-17 | MADHAV | DATABASE QUERY | CRUDE QUERIES | VALID QUERY RESPONSE | VALID QUERY RESPONSE | PASS |
| T5 | 20-09-17 | MADHAV | DATABASE TEARDOWN | DATABASE | DATABASE DESTROYED | DATABASE DESTROYED | PASS |
| T6 | 28-09-17 | UJJVAL | ADMIN LOGIN | ID PASSWORD | PRIVILEDGED LOG IN | PRIVILEDGED LOGIN | PASS |
| T7 | 04-10-17 | DUSHYANT | USER LOGIN | EMAIL PASSWORD | HTTP 200 RESPONSE | ERROR 401: UNAUTHORIZED ACCESS | FAIL |
| T8 | 13-10-17 | DUSHYANT | USER BOOKING | SERVICE, DATE, TIME | SUCCESS MESSAGE | SUCCESS MESSAGE | PASS |
| T9 | 18-10-17 | MADHAV | USER VIEW | DATE, TIME | UI AVAILABLE SEATS | UI AVAILABLE SEATS | PASS |
| T10 | 22-10-17 | UJJVAL | USERS BOOKING | LOGGED INTO ACCOUNT | GENERATE PREVIOUS HISTORY | GENERATE PREVIOUS HISTORY | PASS |
| T11 | 24-10-17 | MADHAV | ADMIN VIEW | DATE, TIME | LIST OF ALL BOOKED CUSTOMERS | LIST OF ALL BOOKED CUSTOMERS | PASS |

# Software Design Document

# INTRODUCTION

## 1.1 PURPOSE

This software design document describes the architecture and system design of our web application, Belleza Scheduling WebApp. It shows how the software system will be structured to satisfy the requirements. It is the primary reference for code and development, and therefore, it must contain all the information that the programmer needs to write the code.

## 1.3 SCOPE

The aim of the software is to provide the user with a facility of seeing the current status and booking an appointment for belleza hair saloon in a much accessible and reliable manner. It also provides a link for peer to peer communication. The software has an additional feature of an my history which helps the users see his previous booking history.

## SYSTEM OVERVIEW

Three major functions have been identified that the software will incorporate. They are as follows:

### 2.2.1 Booking an Appointment

Users can book a specific service on a specific time and date.

### 2.2.2 Finding current status

Users can see the current status of the seats booked and services for which they are booked.

### 2.2.3 History

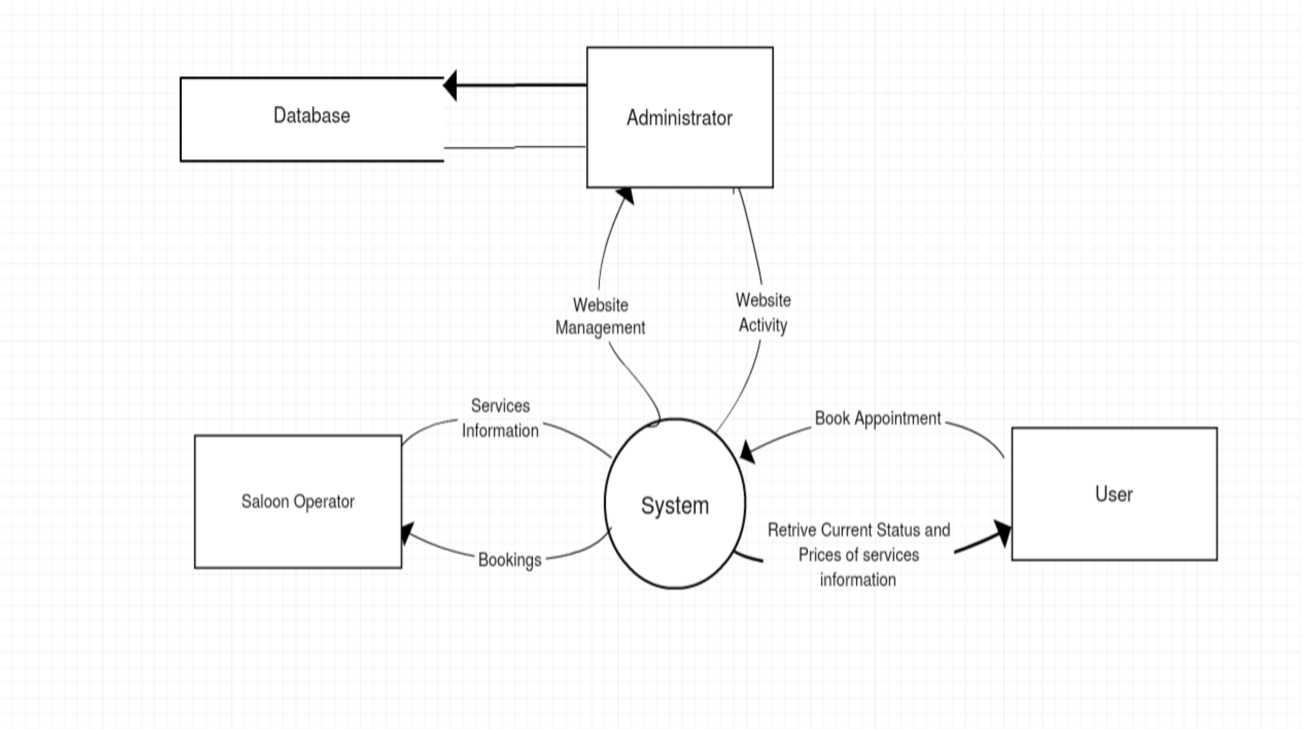
Users can see their previously done bookings through the application.

# 3. SYSTEM ARCHITECTURE

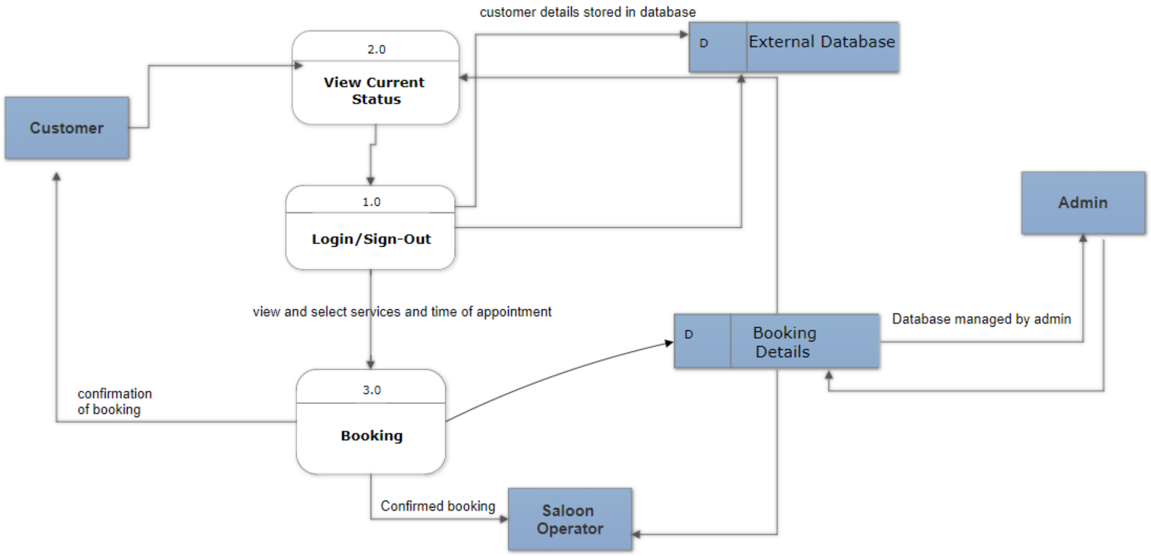
## 3.1 Diagrammatical Representation

This is a high-level view of how functions of the system were allocated to the smaller subsystems.

LEVEL 0



Level 1


customer details stored in database
D
External Database
2.0
View Current Status
Customer
Admin
1.0
Login/Sign-Out
view and select services and time of appointment
D
Booking Details
Database managed by admin
confirmation of booking
3.0
Booking
Confirmed booking
Saloon Operator

# 4. DATA DESIGN

## 4.1 DATA DESCRIPTION

- The 1st input is the SNU G+ Login Credentials. They are not stored in the database, they are tallied with respect to the Google service.
- The second data entity that would be used is tables in SQLite /Firebase; These tables can be created, edited, and cleared with every other Request or Share. The Requests table are refreshed weekly, unless the user wishes to keep the request longer.

## 4.2 DATA DICTIONARY

Alphabetically list the system entities or major data along with their types and descriptions.

Find tab

Share tab

Request tab

Feed

Side menu

G+

Digital Media Marketplace

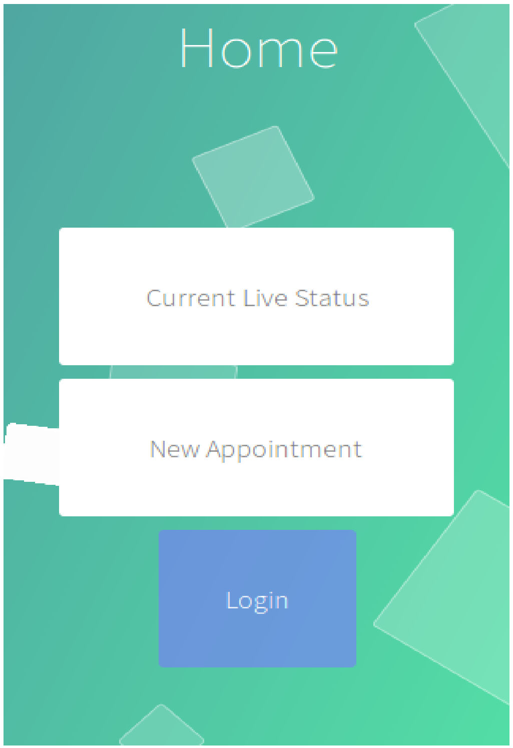
Welcome Screen

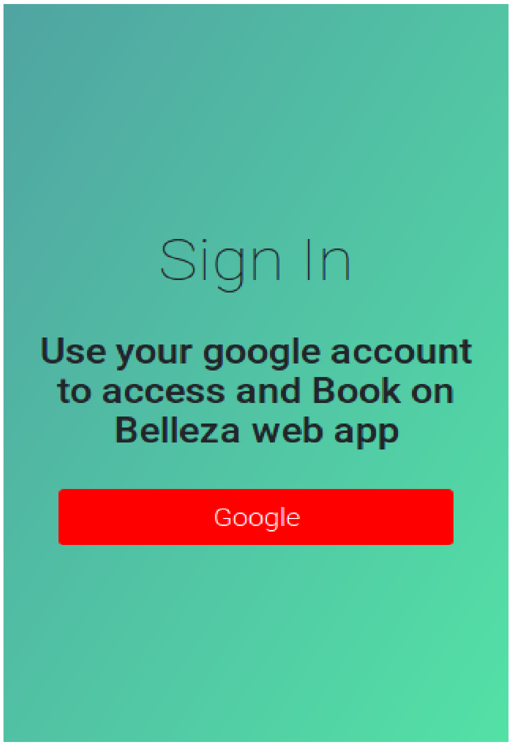
Main Screen

3-line icon

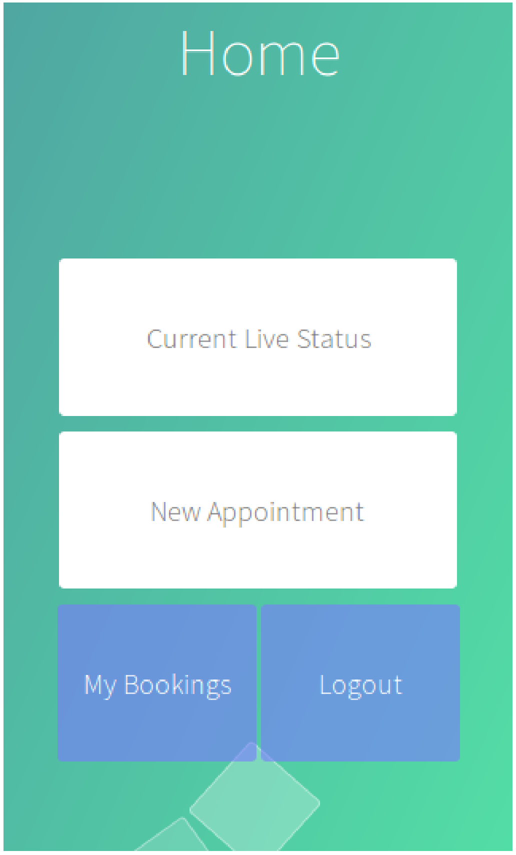
User dashboard

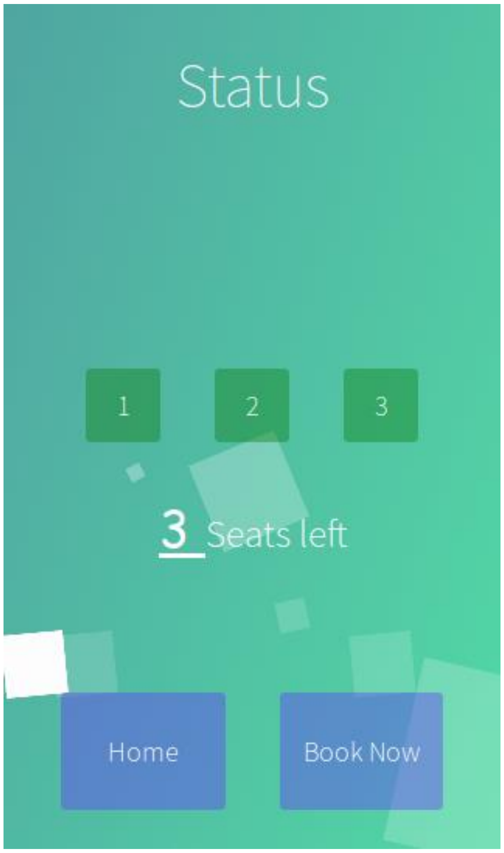
# 5. HUMAN INTERFACE DESIGN

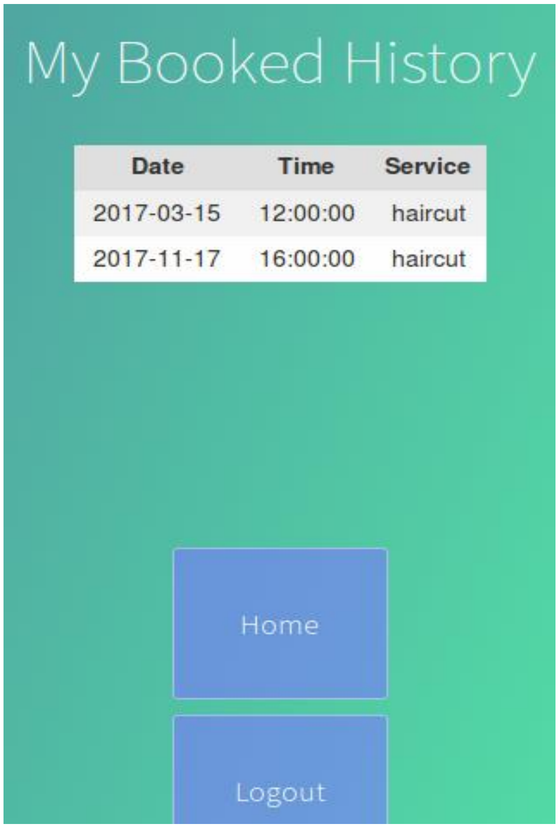




Use your google account to access and Book on Belleza web app

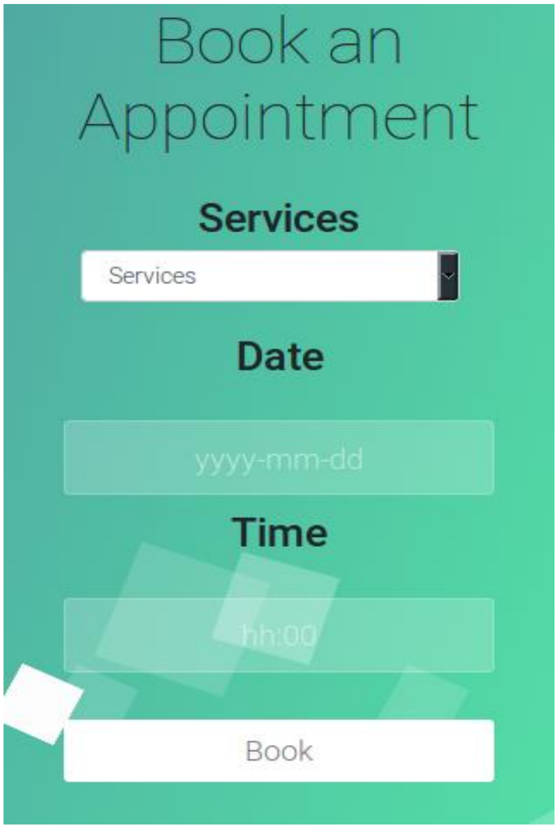






My Booked History

| Date | Time | Service |
|---|---|---|
| 2017-03-15 | 12:00:00 | haircut |
| 2017-11-17 | 16:00:00 | haircut |

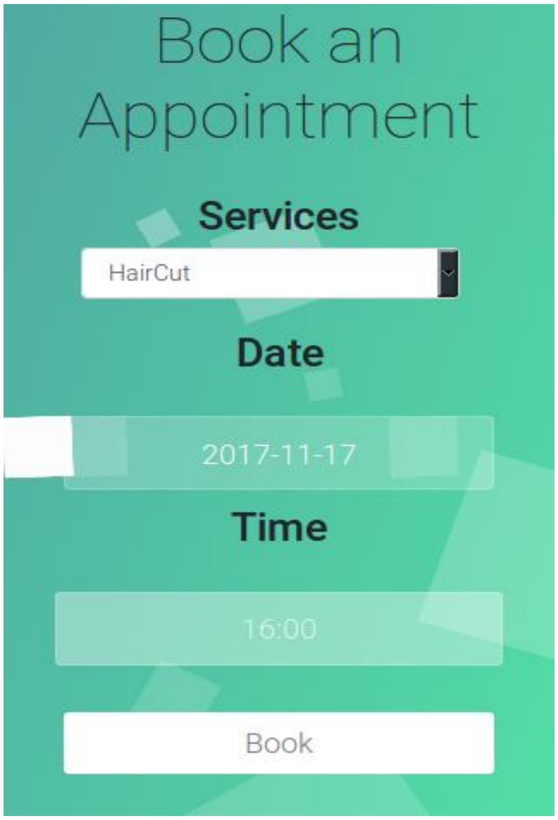


Book an Appointment

Services: Services

Date: yyyy-mm-dd

Time: hh:00

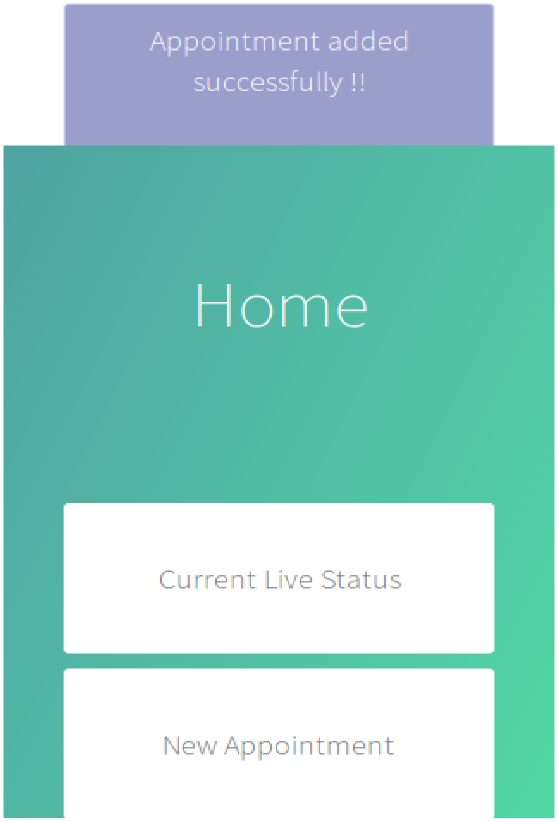


Book an Appointment

Services: HairCut

Date: 2017-11-17

Time: 16:00



Appointment added successfully !!

Home

# 6. REQUIREMENTS MATRIX

| **ID** | **Functional Requirement** | **Technical Specification** | **Implemented in Activity** |
|---|---|---|---|
| **001** | User presses “Username” | Text-field for entering username activated | Welcome Screen |
| **002** | User presses “Password” | Text-field for entering password activated | Welcome Screen |
| **003** | User presses “Submit” | Check in G+ for Authentication | Welcome Screen |
| **004** | User presses “Logout” Button | Deletes user's info from local cache | Main Menu |
| **005** | User presses “Current Live Status” button | Opens " Current Live Status " to display all the current bookings | Home Screen |
| **006** | User Presses "Book Now" Button | Opens Booking screen | Current Live Status Screen |
| **007** | User selects Service | Opens drop down menu of pre-curated categories | Booking Screen |
| **008** | User selects Time | Activates text field for entering details | Booking Screen |
| **008** | User selects Date | Activates text field for entering details | Booking Screen |
| **009** | User taps check box for terms and conditions | Confirms participation in T&Cs | Booking Screen |
| **010** | User taps Submit | Uploads the above details to database | Booking Screen |

| | | | |
|---|---|---|---|
| **011** | User Presses "My Bookings" button | Opens History screen | Main Screen/ Any screen |
| **012** | User Selects date | Filters to bookings on specific date | History Screen |
| **013** | User presses “LOGOUT” button | Logs out the user and opens the home screen | Any Screen |

# Coding Standards

# GENERAL LAYOUT

## INDENTATION:

- Real spaces.
- No tabs.
- No exceptions.
- Maximum line length:
  - ➢ 79 characters with a soft limit for 84 if absolutely necessary. Try to avoid too nested code by cleverly placing break, continue and return statements.
- Continuing long statements:
  - ➢ To continue a statement you can use backslashes in which case you should align the next line with the last dot or equal sign, or indent four spaces:

## EXPRESSIONS AND STATEMENTS:

### GENERAL WHITESPACE RULES:

- No whitespace for unary operators that are not words (e.g.: -, ~ etc.) as well on the inner side of parentheses.
- Whitespace is placed between binary operators.

### BLANK LINES:

- Top level functions and classes are separated by two lines, everything else by one. Do not use too many blank lines to separate logical segments in code.

## COMPARISONS:

- Against arbitrary types: == and !=
- Against singletons with is and is not (eg: foo is not None)
- Never compare something with True or False (for example never do foo ==False, do not foo instead)

## NEGATED CONTAINMENT CHECKS:

- Use foo not in bar instead of not foo in bar.
- Instance checks:
  - isinstance(a, C) instead of type(A) is C, but try to avoid instance checks in general. Check for features.

## NAMING CONVENTIONS:

- Class names: `CamelCase`, with acronyms kept uppercase (`HTTPWriter` and not HttpWriter)
- Variable names: `lowercase_with_underscores`
- Method and function names: `lowercase_with_underscores`
- Constants: `UPPERCASE_WITH_UNDERSCORES`
- precompiled regular expressions: `name_re`

Protected members are prefixed with a single underscore. Double underscores are reserved for mixin classes.

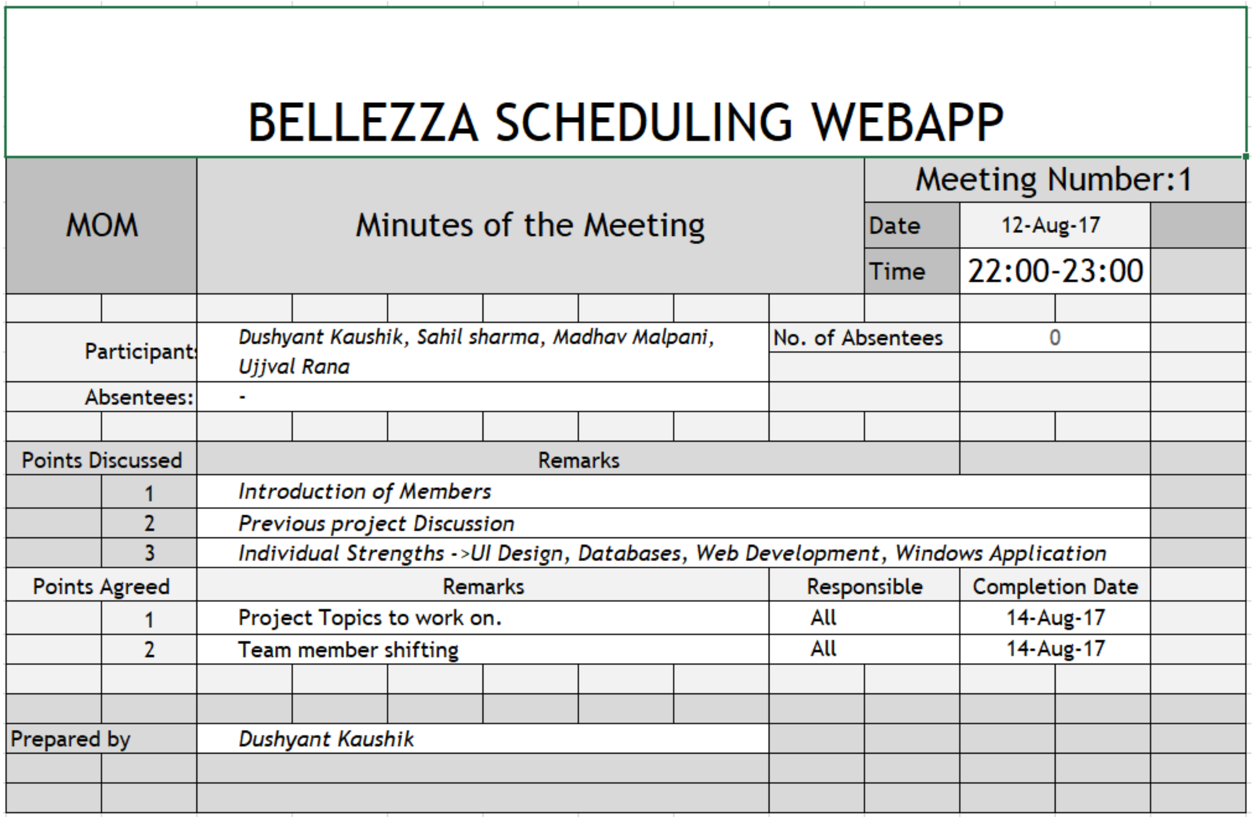
On classes with keywords, trailing underscores are appended. Clashes with builtins are allowed and must not be resolved by appending an underline to the variable name. If the function needs to access a shadowed builtin, rebind the builtin to a different name instead.

### FUNCTION AND METHOD ARGUMENTS:

- class methods: `cls` as first parameter
- instance methods: `self` as first parameter
- lambdas for properties might have the first parameter replaced with x like in `display_name = property(lambda x: x.real_name or x.username)`

# Minutes of Meeting

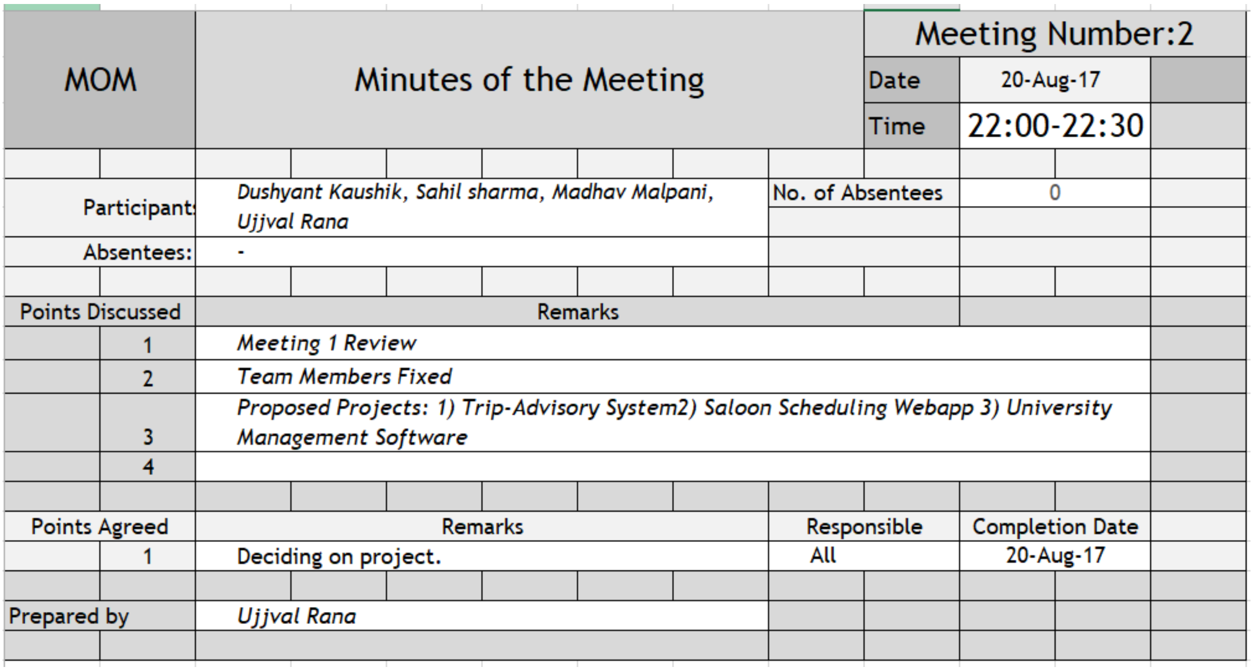
## MEETING 1



BELLEZZA SCHEDULING WEBAPP

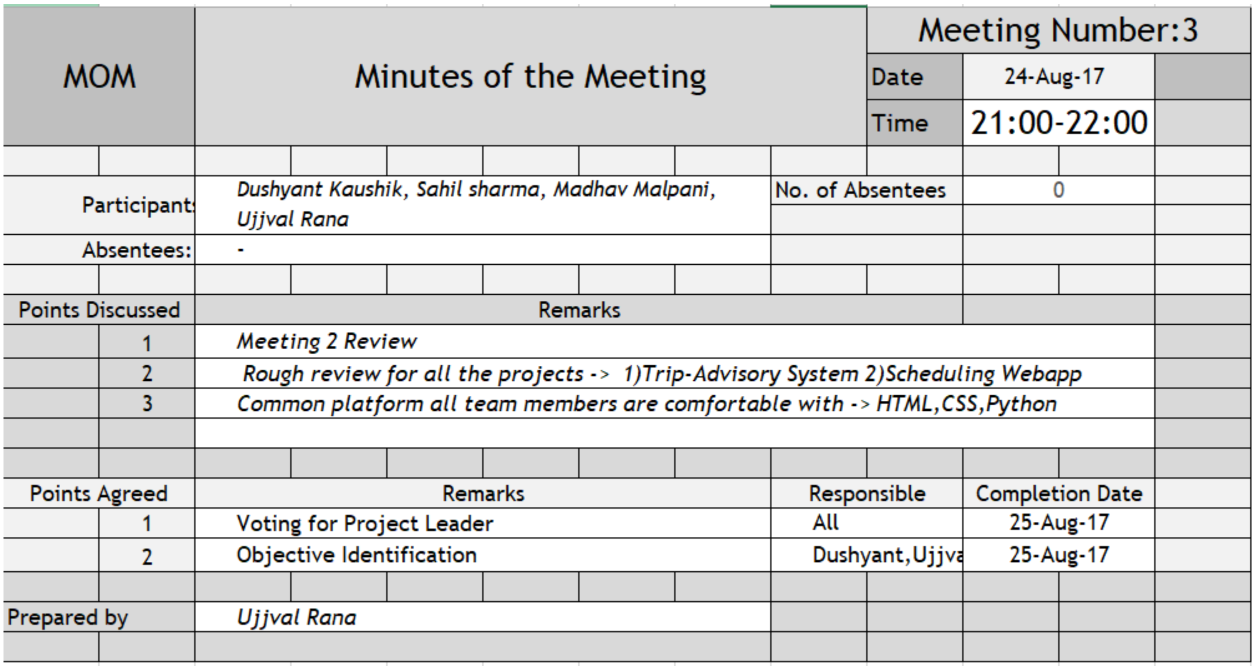
| | | | | |
|---|---|---|---|---|
| MOM | | Minutes of the Meeting | Meeting Number:1 | |
| | | | Date | 12-Aug-17 |
| | | | Time | 22:00-23:00 |
| Participant | | *Dushyant Kaushik, Sahil sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 |
| Absentees: | | - | | |
| Points Discussed | | Remarks | | |
| | 1 | *Introduction of Members* | | |
| | 2 | *Previous project Discussion* | | |
| | 3 | *Individual Strengths ->UI Design, Databases, Web Development, Windows Application* | | |
| Points Agreed | | Remarks | Responsible | Completion Date |
| | 1 | Project Topics to work on. | All | 14-Aug-17 |
| | 2 | Team member shifting | All | 14-Aug-17 |
| Prepared by | | *Dushyant Kaushik* | | |

MEETING 2



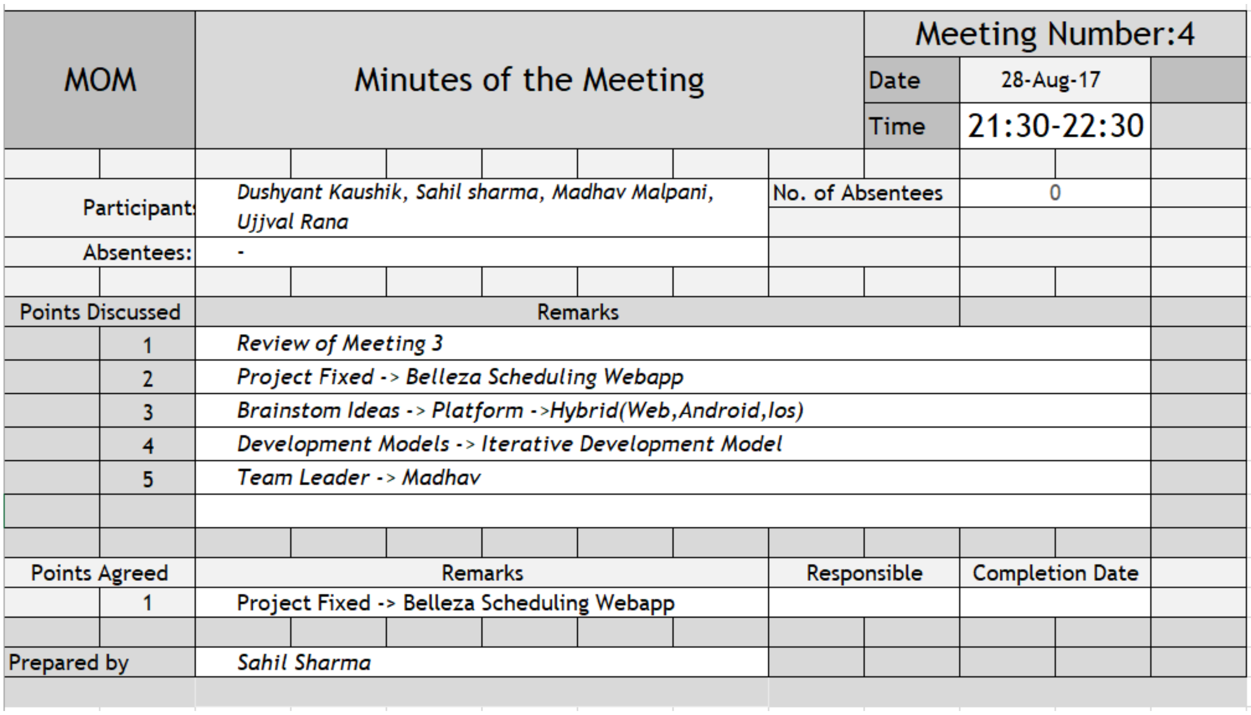
| | | | |
|---|---|---|---|
| MOM | Minutes of the Meeting | Meeting Number:2 | |
| | | Date | 20-Aug-17 |
| | | Time | 22:00-22:30 |
| Participants | *Dushyant Kaushik, Sahil sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 |
| Absentees: | - | | |
| Points Discussed | Remarks | | |
| 1 | *Meeting 1 Review* | | |
| 2 | *Team Members Fixed* | | |
| 3 | *Proposed Projects: 1) Trip-Advisory System2) Saloon Scheduling Webapp 3) University Management Software* | | |
| 4 | | | |
| Points Agreed | Remarks | Responsible | Completion Date |
| 1 | Deciding on project. | All | 20-Aug-17 |
| Prepared by | *Ujjval Rana* | | |

# MEETING 3



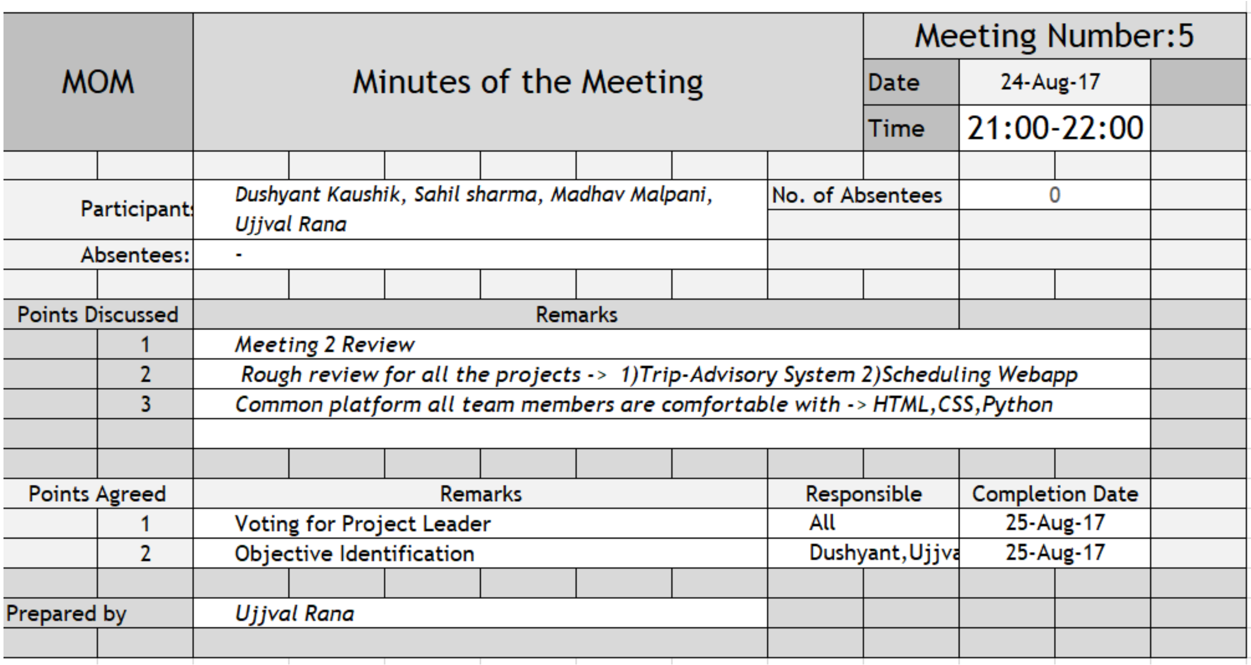
| | | | | |
|---|---|---|---|---|
| MOM | | Minutes of the Meeting | Meeting Number:3 | |
| | | | Date | 24-Aug-17 |
| | | | Time | 21:00-22:00 |
| Participant | | *Dushyant Kaushik, Sahil sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 |
| Absentees: | | - | | |
| Points Discussed | | Remarks | | |
| | 1 | *Meeting 2 Review* | | |
| | 2 | *Rough review for all the projects -> 1)Trip-Advisory System 2)Scheduling Webapp* | | |
| | 3 | *Common platform all team members are comfortable with -> HTML,CSS,Python* | | |
| Points Agreed | | Remarks | Responsible | Completion Date |
| | 1 | Voting for Project Leader | All | 25-Aug-17 |
| | 2 | Objective Identification | Dushyant,Ujjva | 25-Aug-17 |
| Prepared by | | *Ujjval Rana* | | |

# MEETING 4



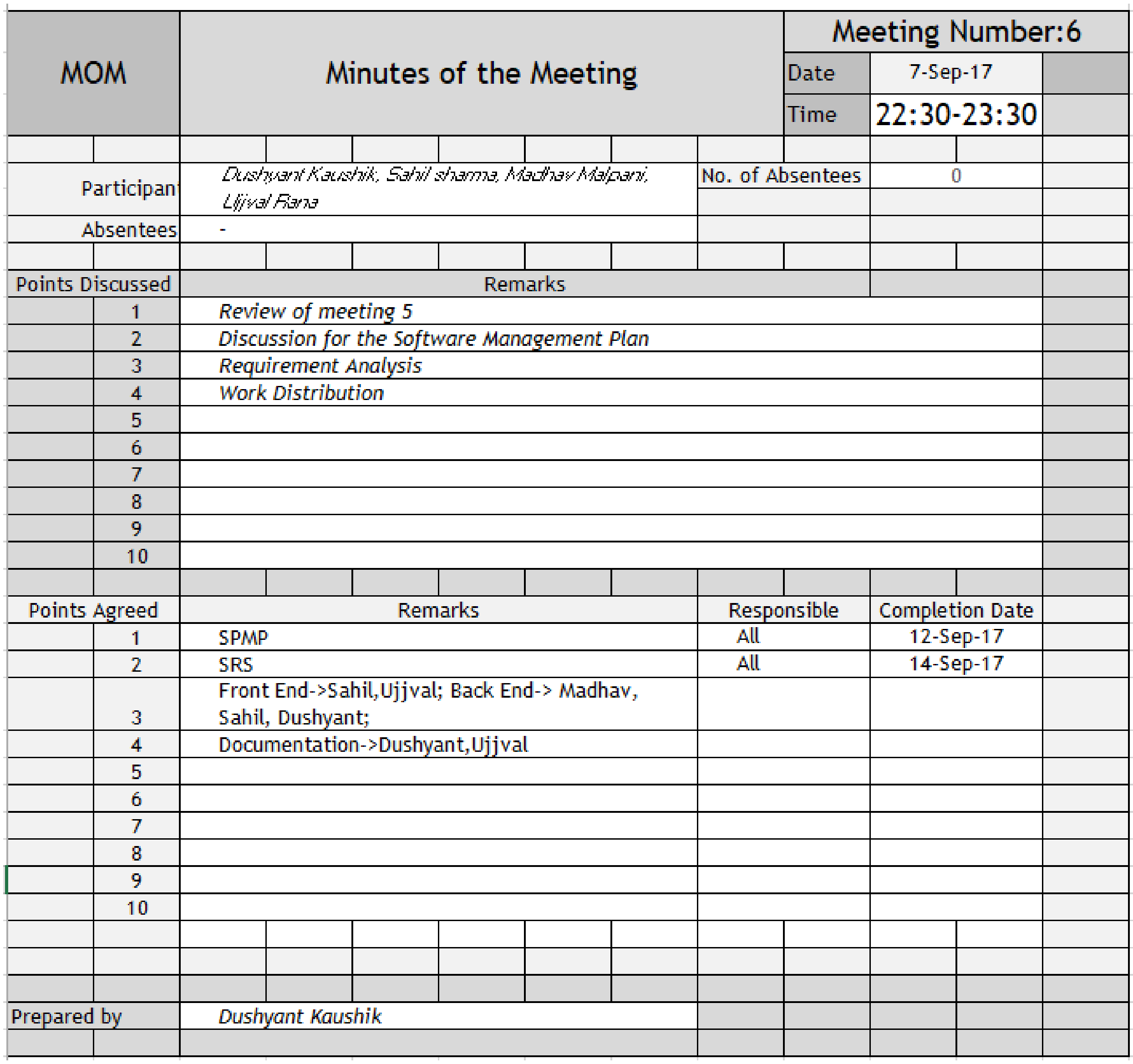
| | | | | | |
|---|---|---|---|---|---|
| MOM | | Minutes of the Meeting | Meeting Number:4 | | |
| | | | Date | 28-Aug-17 | |
| | | | Time | 21:30-22:30 | |
| Participant | | *Dushyant Kaushik, Sahil sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 | |
| Absentees: | | - | | | |
| Points Discussed | | Remarks | | | |
| | 1 | *Review of Meeting 3* | | | |
| | 2 | *Project Fixed -> Belleza Scheduling Webapp* | | | |
| | 3 | *Brainstom Ideas -> Platform ->Hybrid(Web,Android,Ios)* | | | |
| | 4 | *Development Models -> Iterative Development Model* | | | |
| | 5 | *Team Leader -> Madhav* | | | |
| Points Agreed | | Remarks | Responsible | Completion Date | |
| | 1 | Project Fixed -> Belleza Scheduling Webapp | | | |
| Prepared by | | *Sahil Sharma* | | | |

# MEETING 5



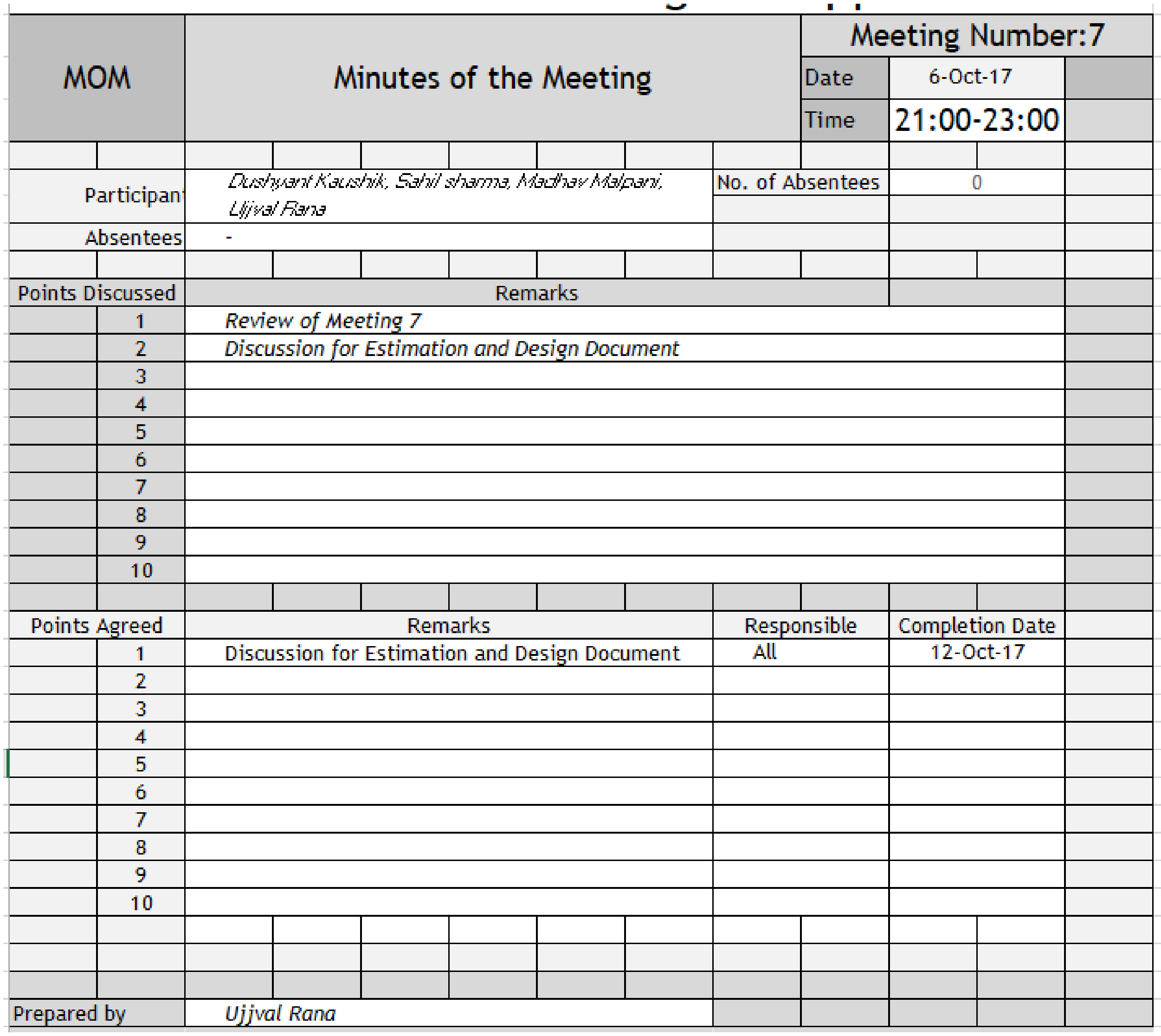
| | | | | |
|---|---|---|---|---|
| MOM | | Minutes of the Meeting | Meeting Number:5 | |
| | | | Date | 24-Aug-17 |
| | | | Time | 21:00-22:00 |
| Participant | | *Dushyant Kaushik, Sahil sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 |
| Absentees: | | - | | |
| Points Discussed | | Remarks | | |
| | 1 | *Meeting 2 Review* | | |
| | 2 | *Rough review for all the projects -> 1)Trip-Advisory System 2)Scheduling Webapp* | | |
| | 3 | *Common platform all team members are comfortable with -> HTML,CSS,Python* | | |
| Points Agreed | | Remarks | Responsible | Completion Date |
| | 1 | Voting for Project Leader | All | 25-Aug-17 |
| | 2 | Objective Identification | Dushyant,Ujjva | 25-Aug-17 |
| Prepared by | | *Ujjval Rana* | | |

MEETING 6



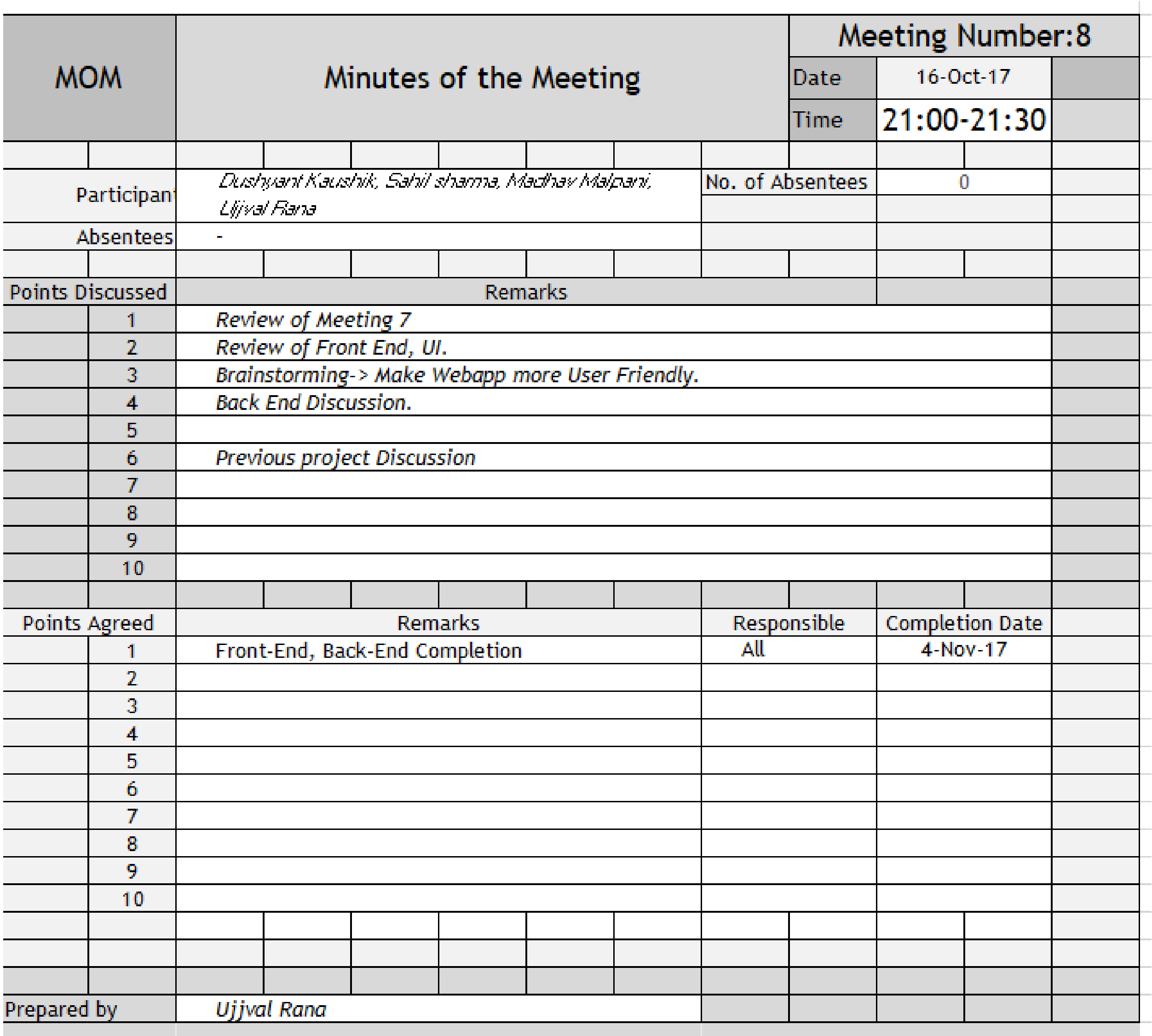
| | | | | |
|---|---|---|---|---|
| MOM | | Minutes of the Meeting | Meeting Number:6 | |
| | | | Date | 7-Sep-17 |
| | | | Time | 22:30-23:30 |
| Participan | | *Dushyant Kaushik, Sahil sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 |
| Absentees | | - | | |
| Points Discussed | | Remarks | | |
| | 1 | *Review of meeting 5* | | |
| | 2 | *Discussion for the Software Management Plan* | | |
| | 3 | *Requirement Analysis* | | |
| | 4 | *Work Distribution* | | |
| | 5 | | | |
| | 6 | | | |
| | 7 | | | |
| | 8 | | | |
| | 9 | | | |
| | 10 | | | |
| Points Agreed | | Remarks | Responsible | Completion Date |
| | 1 | SPMP | All | 12-Sep-17 |
| | 2 | SRS | All | 14-Sep-17 |
| | 3 | Front End->Sahil,Ujjval; Back End-> Madhav, Sahil, Dushyant; | | |
| | 4 | Documentation->Dushyant,Ujjval | | |
| | 5 | | | |
| | 6 | | | |
| | 7 | | | |
| | 8 | | | |
| | 9 | | | |
| | 10 | | | |
| Prepared by | | *Dushyant Kaushik* | | |

# MEETING 7



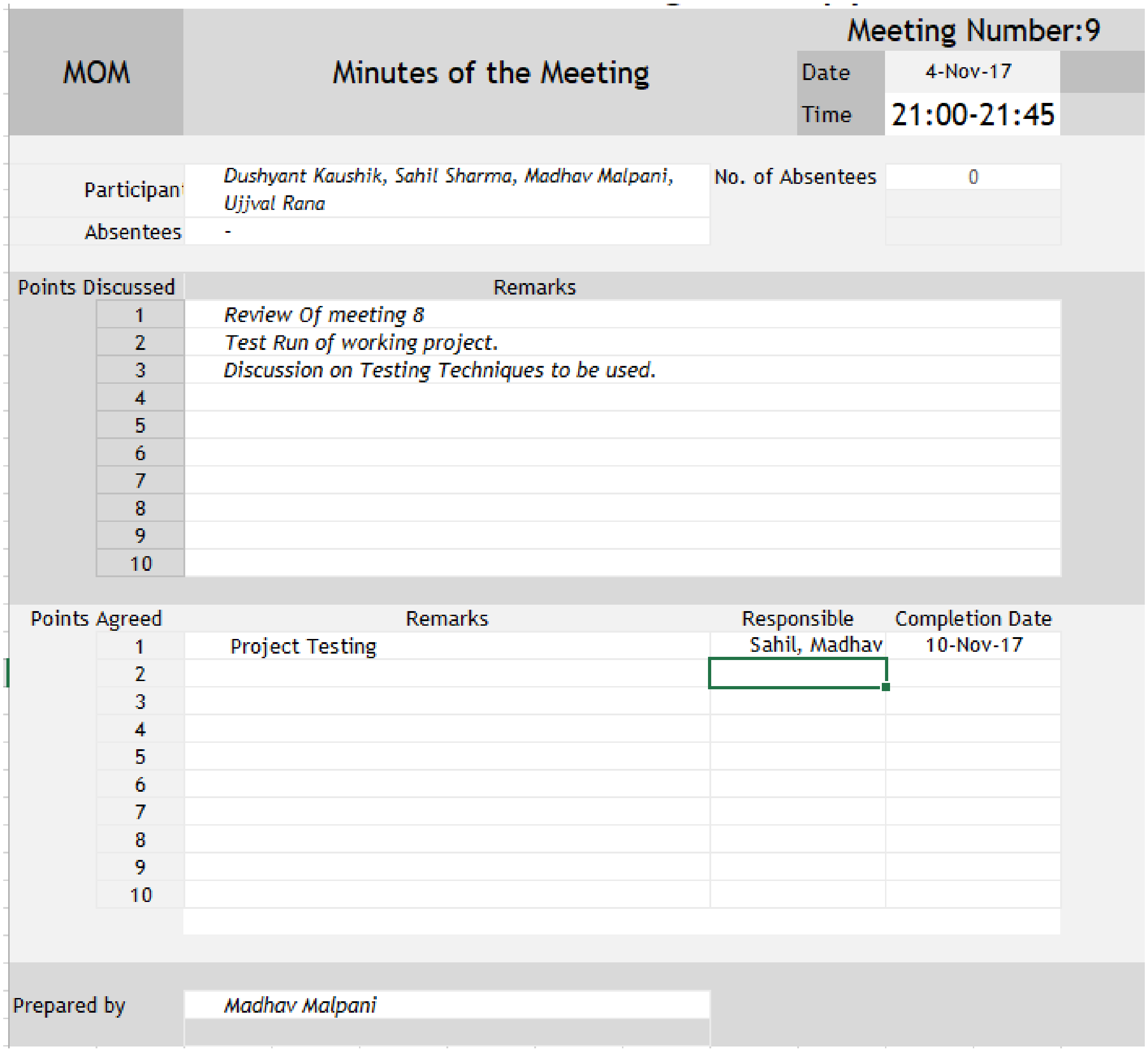
| | | | | | |
|---|---|---|---|---|---|
| MOM | | Minutes of the Meeting | Meeting Number:7 | | |
| | | | Date | 6-Oct-17 | |
| | | | Time | 21:00-23:00 | |
| Participan | | *Dushyant Kaushik, Sahil sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 | |
| Absentees | | - | | | |
| Points Discussed | | Remarks | | | |
| | 1 | *Review of Meeting 7* | | | |
| | 2 | *Discussion for Estimation and Design Document* | | | |
| | 3 | | | | |
| | 4 | | | | |
| | 5 | | | | |
| | 6 | | | | |
| | 7 | | | | |
| | 8 | | | | |
| | 9 | | | | |
| | 10 | | | | |
| Points Agreed | | Remarks | Responsible | Completion Date | |
| | 1 | Discussion for Estimation and Design Document | All | 12-Oct-17 | |
| | 2 | | | | |
| | 3 | | | | |
| | 4 | | | | |
| | 5 | | | | |
| | 6 | | | | |
| | 7 | | | | |
| | 8 | | | | |
| | 9 | | | | |
| | 10 | | | | |
| Prepared by | | *Ujjval Rana* | | | |

# MEETING 8



| | | | | |
|---|---|---|---|---|
| MOM | | Minutes of the Meeting | Meeting Number:8 | |
| | | | Date | 16-Oct-17 |
| | | | Time | 21:00-21:30 |
| Participant | | *Dushyant Kaushik, Sahil sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 |
| Absentees | | - | | |
| Points Discussed | | Remarks | | |
| | 1 | *Review of Meeting 7* | | |
| | 2 | *Review of Front End, UI.* | | |
| | 3 | *Brainstorming-> Make Webapp more User Friendly.* | | |
| | 4 | *Back End Discussion.* | | |
| | 5 | | | |
| | 6 | *Previous project Discussion* | | |
| | 7 | | | |
| | 8 | | | |
| | 9 | | | |
| | 10 | | | |
| Points Agreed | | Remarks | Responsible | Completion Date |
| | 1 | Front-End, Back-End Completion | All | 4-Nov-17 |
| | 2 | | | |
| | 3 | | | |
| | 4 | | | |
| | 5 | | | |
| | 6 | | | |
| | 7 | | | |
| | 8 | | | |
| | 9 | | | |
| | 10 | | | |
| Prepared by | | *Ujjval Rana* | | |

# MEETING 9



| MOM | Minutes of the Meeting | Meeting Number:9 | |
|---|---|---|---|
| | | Date | 4-Nov-17 |
| | | Time | 21:00-21:45 |

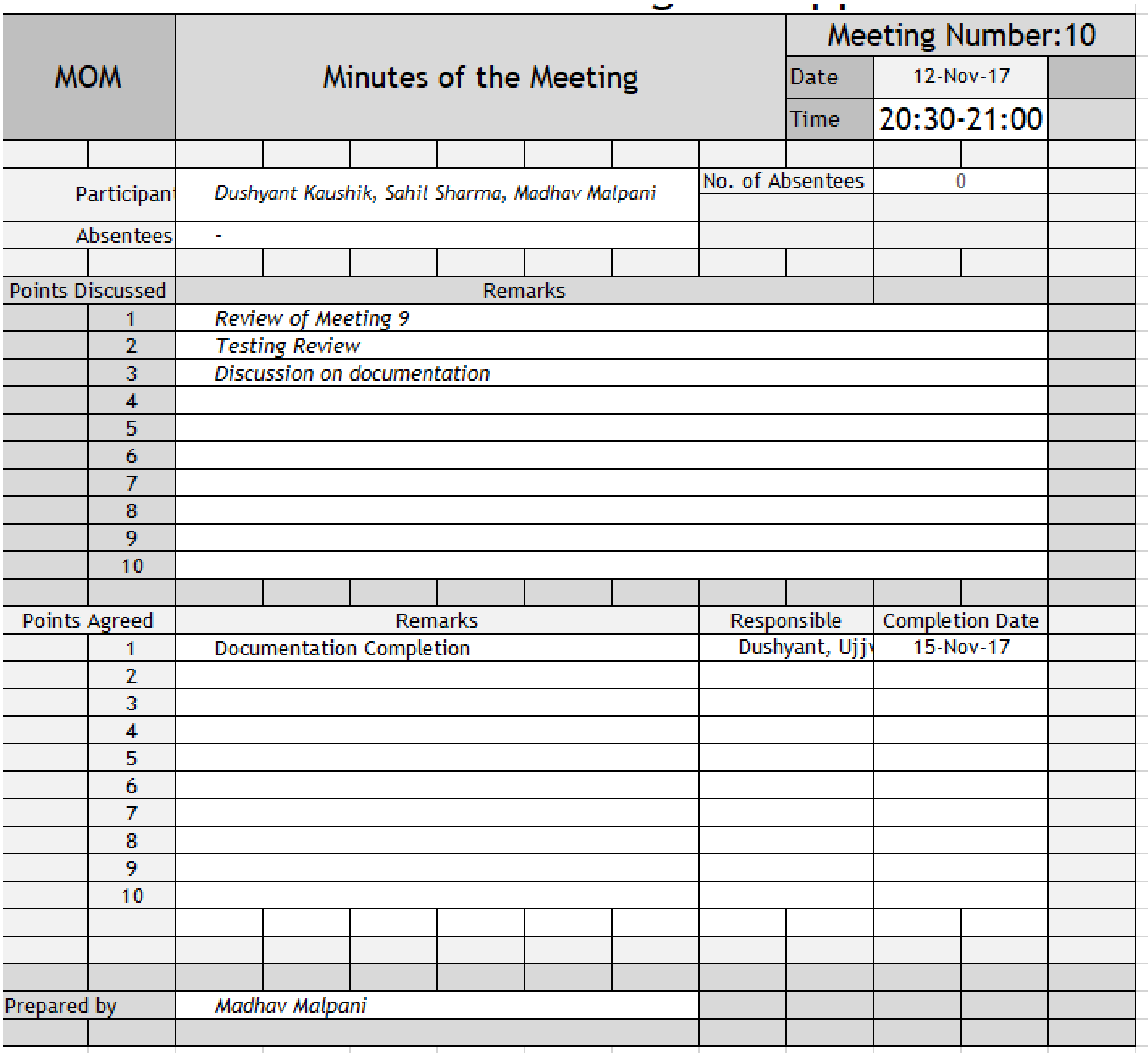
| | | | |
|---|---|---|---|
| Participan | *Dushyant Kaushik, Sahil Sharma, Madhav Malpani, Ujjval Rana* | No. of Absentees | 0 |
| Absentees | - | | |

| Points Discussed | Remarks |
|---|---|
| 1 | *Review Of meeting 8* |
| 2 | *Test Run of working project.* |
| 3 | *Discussion on Testing Techniques to be used.* |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |

| Points Agreed | Remarks | Responsible | Completion Date |
|---|---|---|---|
| 1 | Project Testing | Sahil, Madhav | 10-Nov-17 |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |

| | |
|---|---|
| Prepared by | *Madhav Malpani* |

MEETING 10



| | | | |
|---|---|---|---|
| MOM | Minutes of the Meeting | Meeting Number:10 | |
| | | Date | 12-Nov-17 |
| | | Time | 20:30-21:00 |

| | | | |
|---|---|---|---|
| Participan | *Dushyant Kaushik, Sahil Sharma, Madhav Malpani* | No. of Absentees | 0 |
| Absentees | - | | |

| Points Discussed | | Remarks |
|---|---|---|
| | 1 | *Review of Meeting 9* |
| | 2 | *Testing Review* |
| | 3 | *Discussion on documentation* |
| | 4 | |
| | 5 | |
| | 6 | |
| | 7 | |
| | 8 | |
| | 9 | |
| | 10 | |

| Points Agreed | | Remarks | Responsible | Completion Date |
|---|---|---|---|---|
| | 1 | Documentation Completion | Dushyant, Ujj | 15-Nov-17 |
| | 2 | | | |
| | 3 | | | |
| | 4 | | | |
| | 5 | | | |
| | 6 | | | |
| | 7 | | | |
| | 8 | | | |
| | 9 | | | |
| | 10 | | | |

| | |
|---|---|
| Prepared by | *Madhav Malpani* |

**THANK YOU**

**END OF DOCUMENT**