March 2016

**Version 1.0**

**Anand**

**Functional Specification**

**Appointment Booking System**

**DocHelper**

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Description | Author | Reviewed by |
| 2016/03/01 | 1.0 | Initial draft | Anoop G |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Approval History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Approved by | Remarks |
| 2016/03/07 | 1.0 |  | Base version |
|  |  |  |  |
|  |  |  |  |

**Table of contents -**

[1 Introduction 4](#_Toc445125868)

[1.1. Purpose 4](#_Toc445125869)

[1.2. Abbreviations 4](#_Toc445125870)

[1.3. Terminology 4](#_Toc445125871)

[2 System overview 4](#_Toc445125872)

[2.1. Objectives 4](#_Toc445125873)

[2.2. Block diagram 5](#_Toc445125874)

[2.3. System users 5](#_Toc445125875)

[2.4. Resource availability 5](#_Toc445125876)

[2.5. Priority booking 6](#_Toc445125877)

[2.5.1. Hot Slot/ Tatkal Slot 6](#_Toc445125878)

[2.5.2. Weightage 7](#_Toc445125879)

[2.5.3. Auctions 7](#_Toc445125880)

[2.6. Unreserved Appointments or direct appointments 8](#_Toc445125881)

[2.7. Appointment cancellation 9](#_Toc445125882)

[2.8. Consultation Sequence 9](#_Toc445125883)

[2.9. Emergency handling 9](#_Toc445125884)

[2.10. Booking Process (from Web) 10](#_Toc445125885)

[3 System Requirements 10](#_Toc445125886)

[4 Functional requirements 12](#_Toc445125887)

[3.1. Refresh process 12](#_Toc445125888)

[3.2. System configuration 12](#_Toc445125889)

[3.3. Create appointment 12](#_Toc445125890)

[3.4. Start Consultation 13](#_Toc445125891)

[3.5. Hold Consultation 13](#_Toc445125892)

[3.6. Delete Appointment 13](#_Toc445125893)

[1.1. Status display 14](#_Toc445125894)

[1.2. Communication 14](#_Toc445125895)

[1.3. Error and failure handling 14](#_Toc445125896)

[1.4. Logging 14](#_Toc445125897)

[1.5. View and export logs 15](#_Toc445125898)

[1.6. Multi language support 15](#_Toc445125899)

[1.7. UI animations 15](#_Toc445125900)

[5 Non-functional requirements 15](#_Toc445125901)

[4.1. Design considerations 15](#_Toc445125902)

[6 Configuration parameters 15](#_Toc445125903)

[1.1. System configuration 16](#_Toc445125904)

[1.2. Operational parameters 16](#_Toc445125905)

[7 Communication interface 16](#_Toc445125906)

[6.1. Attributes 16](#_Toc445125907)

[6.2. List of Web APIs 16](#_Toc445125908)

[6.3. API Parameters 17](#_Toc445125909)

[6.3.1. Data types 17](#_Toc445125910)

[8 User interface 17](#_Toc445125911)

[7.1. Salient features 17](#_Toc445125912)

[7.2. Status messages 18](#_Toc445125913)

[9 Appendices 18](#_Toc445125914)

[1.1 Status code categories 18](#_Toc445125915)

[1.2 Status codes 18](#_Toc445125916)

[10 References 19](#_Toc445125917)

# Introduction

## Purpose

This document serves as the *Functional Specification* for the　Appointment Booking System. Following are the primary objectives:

* As an input to the development team for detailed design.
* As an input to the QA team for system and acceptance test cases.
* As an input to product owners and other stake holders to understand what they would be getting.

## Abbreviations

|  |  |
| --- | --- |
| Appt | Appointment |
| GUI | Graphical User Interface |
| HTTP | Hyper Text Transfer Protocol |
| OS | Operating System |
| QA | Quality Assurance |
| RFC | Request For Comments |
| UI | User Interface |
| RA | Resource Administrator |
|  |  |

## Terminology

|  |  |
| --- | --- |
| API Server | The web server that services certain requests from the System |
| Refresh process | The sequence of steps which System executes to obtain a set of operational parameters from the API server. |
| Appointments | Same as meeting/reservation/ Booking. |
| Booking Terminal | The terminal device near each doctor’s cabin. |
| Doctor’s View | Web based system on each doctor’s Tablet/Laptop. |

# System overview

## Objectives

The Appointment Booking System is the GUI interface of booking appointments with doctors. The system is primarily used for managing appointments automatically based on the availability of resources. System contains single or multiple resources/doctors.

Main sections in this Web application are given below:

1. Service to manage resources, appointments based on resource availability.
2. Doctor’s view=> Service to view current day appointments as tokens, in tabular form.
3. Booking Terminal=> Service to view ongoing and immediate future appointments of the day.
4. Reception Terminal=> Service to display current day appointments with sliding effect. Patients can watch their token position and the available free slots.

## Block diagram

Terminal

Reception

Database

Website

API Service

Web

Data is shared between Web server and API server

API call / API Response (http/https/jSon)

Http request/response (Html/JavaScript)

Doctor’s view

Figure 1 Overview

## System users

* Administrator => Admin user can add edit resources, users etc...
* RA =>RA can modify resource availability and other resource settings.
* Bookers => Bookers can update their profiles and add new

Appointments, update own appointments.

## Resource availability

RA can set resource availability for every day of a week throughout a period. It can be a year/single month/multiple months etc… Availability slots can be set by the RA. Default availability slot/ usual availability slot can be set by RA, but it is always modifiable as a whole or in partial. For example RA can set for all Mondays, all alternate Mondays, all Thursday and Friday, etc…

Sample below:

Default time slot is 15:00 to 21:00

RA can modify/override this by own availability on a day.

For example

16:00 to 19:00 for Mondays.

14:00 to 18:00 for Tuesday and Thursday.

Etc…

## Priority booking

RA can set some slot as priority slot based on the higher number of appointments, the patient’s age and distance.

## Hot Slot/ Tatkal Slot

Hot slot is a consulting slot with higher consultation fee. RA can select one or more slots as hot slot. Hot slots can be enabled or disabled from resource settings as a numeric number.

For example:

-1=> indicates hot slot is disabled.

0=> indicates slot booking available today.

1=> indicates slot booking available from one day before.

n=> indicates slot booking available from n days before.



Figure 2 Hot Slot

## Weightage

Patients can be given weightage based on age and distance. Those patients can come on a particular slot set by the RA. These selected slots will not be available for other users till RA permits from the settings. RA can set one or more slots in weightage.

Sample data below:

Weightage enabled : true.

Visibility Period : numeric, N

Distance => 50 KM

Patient age <=2 & >=60

Weightage slots will be visible for all users’ N days ahead from the consultation day. 

Figure 3 Weightage

## Auctions

A patient who could not get his preferred slot and hence booked some/any available slot can participate slot auction. Along with current booking, this patient can request his preference with an extra amount. RA can offer/execute appointment based on the higher auction amount. The order of consultation at a time is in the decreasing order of auction amount offered.

For example: Patient-1 booked an available slot, but he can select his preferred time T with an extra amount A1. Patient-2 and Patient-3 do the same with amounts A2 and A3 respectively. Let A2>A1>A3. Then, the order of consultation is Patient-2, Patient-1 and Patient-3.

 **Figure 4 Auction work flow of Bookers**



**Figure 5 Auction work flow @ Doctor’s view**

## Unreserved Appointments or direct appointments

Patients who do not fix an appointment in advance can book from the terminal. These bookings will be added into a separate list with a status *“From Terminal”*. RA can decide upon their consultation time.

## Appointment cancellation

RA can cancel a single appointment, multiple appointments, single slot, multiple slots, a whole day, multiple days, single week, multiple weeks, single month, multiple months etc… RA can notify cancellation along with a reason. All cancellation options are available from the web. Cancellation for the current day is possible from the doctor’s view too.

## Consultation Sequence

RA can arrange auction participants, prior appointments, un-reserved appointments etc. using auto-sort /drag-drop mouse operations.

Appointments will be sorted based on the start-time and sort-order. Absentee will be added into a wait-list. A new booking time will be provided for him based on a counter setting. This will be repeated till the end of the appointment.

Example below: Consider wait-list position counter is 3.

A1

A2

A3

A4

A5

A1 is absent

A2

A3

A4

A1

A5

A2 is absent

A3

A4

A1

A2

A5

**Figure 6 Consultation Sequence**

## Emergency handling

RA may cancel or postpone appointments on following reasons.

1. Emergency appointment
2. Emergency hospital visit
3. Personal reason

RA can display the reason on the terminal.

Example: Current time: 15:00

Hold => 1 hour

All upcoming meetings start and end time should updated as below

15:00 – 15:15

16:00 – 16:15

15:15– 15:30

16:15 – 16:30

15:30 – 15:45

16:30 – 16:45

**Figure 7 Emergency handling-I**

If RA is back to work after 0.5 hour, all meetings’ start and end time will be updated as below,

16:00 – 16:15

16:15 – 16:30

15:30 – 15:45

16:30 – 16:45

15:45 – 16:00

16:00 – 16:15

**Figure 8 Emergency handling-II**

## Booking Process (from Web)

User registration should be done before any reservation. A new user can start the registration using e-mail id and/or mobile phone number. An OTP is preferably sent to the registered mobile phone number. After the successful OTP verification, user will be provided a detailed registration page. Hereafter, the user can book appointments with the doctor using the details from the registration phase.

*[Ref: Excel file ‘Specification-Layout\_v1.0’ => ‘User Registration’](file:/E:/ICS-Development/Specifications/DocHelper/Specification-Layouts.xlsx)*

# System Requirements

The system will be provided as a web application. Since the system is provided as web application, No Separate installation is required in the user PC. A standard authenticated PC with an internet browser will be sufficient to use the system.

**Web server/Database server:** One windows 2008 server which will behave as Web server and database server

**Server Hardware Requirement**

|  |  |
| --- | --- |
| Component | Requirement |
| Processor | Processor type:   * Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support   Processor speed:   * Minimum: 2.0 GHz * Recommended: 2.4 GHz or faster |
| Memory | RAM:   * Minimum:2GB * Recommended: 6 GB or more |
| HDD | Minimum : 200 GB  Recommended : 300GB  Partitions:  C : System Disk – 100 GB  D: Data – 200GB  RAID : RAID3 or RAID 5 |

**Figure 9 Hardware Requirement**

**Server - Software Requirement**

|  |  |
| --- | --- |
| Component | Requirement |
| OS | * Windows 2008 standard R2 64 Bit – English |
| Database | * SQL Server 2008 Standard (64-bit) x64 – English |
| .NET Framework | * .NET Framework 4.5 |
| Web server software | * IIS 7.5 |
| Security | * Server should be in the Company Domain * IIS should be enabled for Windows Authentication |

**Figure 10 Software requirement**

**Client PC Requirement**

|  |  |
| --- | --- |
| Component | Requirement |
| Operating system | Windows XP , Windows 7, Windows Vista, Windows 8 |
| Browser | * IE 10 or above * Latest Chrome browser * Latest Firefox browser * Java script support should be enabled |
| Security | * PC should be in the Company Domain so as to authenticate the application against Active directory using Windows login credentials |

**Figure 11 Client PC requirement**

# Functional requirements

Business logic will be implemented by the API server. The responsibility of the system is limited to presentation of data from server, collection of user inputs and communication with the API server. Major use cases are described in the subsequent sections.

## Refresh process

**Use cases**

1. Terminal services [Doctor’s view, terminal, reception] execute subsequent refresh processes in a predefined interval obtained from operational parameters.
2. System executes refresh process immediately when a popup window is closed regardless of submitting/cancelling the operation after create/delete appointment.

## System configuration

**Use cases**

1. User can set all parameters required to connect to API server.
2. System will collect all initial settings from the cache for each resource.

## Create appointment

**Use cases**

**From Web**

1. Booker can create a new appointment.
2. Booker can enter name, age and travelling distance of the patient.
3. Booker can select the preferred date from the page.
4. Booker can select start time and end time of appointment from a predefined table.
5. A booker, who could not get his preferred slot, can input auction time if this feature is activated by RA.
6. For multiple resources, booker can search resources based on their name, disease, specification etc…

*[Ref: Excel file ‘Specification-Layout\_v1.0’ => ‘User-View (Single Doctor)’ & ‘User-View (Multiple Doctors)’](file:/E:/ICS-Development/Specifications/DocHelper/Specification-Layouts.xlsx)*

**From Terminal**

1. Booker can create a new appointment.
2. Booker can enter their mobile number.
3. An invalid/not registered booker can enter their details for booking.
4. Booker can select any available free slot.
5. If free slot is not available, system will display a read only start time which is the end time of last meeting.

*[Ref: Excel file ‘Specification-Layout\_v1.0’ => ‘Terminal View’](file:/E:/ICS-Development/Specifications/DocHelper/Specification-Layouts.xlsx)*

## Start Consultation

**Use cases**

1. System (doctor’s view) will display all available bookings.
2. RA can view whether the auctions are available or not.
3. RA can adjust the bookings by drag and drop. He can sort the appointments here based on auction amount.
4. RA can hold splash screens @ terminal and receptions as he wishes.
5. RA can reset the splash screens once the system is ready.
6. RA can start appointments one by one.
7. Terminal and reception screens will display the ongoing and upcoming token numbers.
8. RA can view direct appointments
9. RA can call any appointments from doctor’s view screen.

*[Ref: Excel file ‘Specification-Layout\_v1.0’ => ‘Doctor's-View’](Specification-Layouts.xlsx)*

## Hold Consultation

**Use cases**

1. RA can hold consultation for a predefined duration.
2. Duration can be chosen from multiples of fifteen minutes.
3. RA can select or enter a valid reason and this can be displayed on terminal and reception screens.
4. System will reset start and end time of all upcoming appointments while setting the duration.

## Delete Appointment

1. RA can delete appointments one by one or as a whole from doctor’s view screen.
2. RA can delete one by one, single slot, multiple slot, whole day, multiple days, a week, multiple weeks etc… from the web.
3. System will notify all patients about the cancellation.

## Status display

**Use cases**

1. Booker can view current token number and recent 5 tokens on terminal main screen *[Ref: Excel file ‘Specification-Layout\_v1.0’ => ‘Terminal View’](Specification-Layouts.xlsx)*
2. Reception system will display all available tokens with status, *[Ref: Excel file ‘Specification-Layout\_v1.0’ => ‘Reception View’](file:/E:/ICS-Development/Specifications/DocHelper/Specification-Layouts.xlsx)*.
3. System displays current server time.

## Communication

**Use cases**

1. System supports both http and https protocols. This should be configurable.
2. System calls Web APIs for all interactions with server.
3. System converts Web API parameters to the designated format understandable by the API server.
4. System converts API results back to the format understandable by the system.

## Error and failure handling

**Use cases**

1. System displays error messages obtained from server.
2. System displays messages for all local errors (communication errors etc.)
3. System retries operation for all communication errors including (not limited to) Network down, Server down, API down, unable to connect to server, connection timeout errors. White screen should not be displayed.
4. Connection timeout value should be 1 minute.

## Logging

**Use cases**

1. System logs all error messages locally (on the device) as and when they happen.
2. System logs information messages of refresh process so as to understand the timing of refresh operation.
3. Log messages should preferably be in English.

## View and export logs

**Use cases**

1. User can view all local log messages currently available.
2. User can explicitly export log messages to API server.
3. Log message should be displayed in the following format:

<Seq.No><dot><Message><space><YYYY-MM-DD><Space><hh:mm:ss>

E.g.: 1. All logs cleared 2014-10-20 10:04:35

<Seq.No> is a display only parameter.

## Multi language support

**Use cases**

1. System has the capability to display captions in multiple languages.
2. English is the language supported at present.
3. The current language setting is obtained from operational parameters.
4. User can change the current language from web.

## UI animations

**Use cases**

1. Animated bubbles in the background of main screen. This is configurable.
2. Animated display of popup screens.
3. Animation in display of time in main screen.

# Non-functional requirements

## Design considerations

1. System is expected to run 24 x 7 as long as possible without much performance issues. Typical duration of single execution is 6 months. Self-recovery is expected for in-between failures.
2. Display styles should be defined at a central location in such a way that it can be modified through a single change.
3. Same captions should not be declared more than once.

# Configuration parameters

There are two types of configurations; system configuration and operational parameters. System configurations are used for static configuration and operational parameters are used for dynamic configuration.

## System configuration

These are the values configurable in the settings screen.

|  |  |  |
| --- | --- | --- |
| **Text/Parameter** | **Default value** | **Description** |
| Company code | Empty | Company code. This value is optional. |
| Terminal ID | MAC address of the device without colons | Terminal ID should be device MAC address by default. User can modify this value. |
| API Server | Empty | IP Address/Domain name of the API server. Both supported. |
| Port | 80 for HTTP  443 for HTTPS | TCP/IP Port of the API server |
| Encrypted Key | Empty | Encrypted key for security. Each Terminal should have a valid Encrypted Key |

## Operational parameters

These are the values obtained from API server through refresh process. Default values are to be used before first refresh process or when a particular parameter is not obtained from server for at least once.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Description** | **Type** | **Default value** |
| Language | Language of the room terminal | Enumeration | en |
| ResourceName | Resourcename of the terminal | String |  |
| StartTimeLimit | The minimum start time allowed. | Time |  |
| EndTimeLimit | The maximum end time allowed. | Time |  |

# Communication interface

## Attributes

|  |  |
| --- | --- |
| **Type** | **Value** |
| Protocol | HTTP and HTTPS |
| Charset | UTF-8 |
| Method | POST |
| Content-type | jSON |
| Web API URL | <Protocol>://<Server>[:<Port>]/<Service path>/<API Name>  Examples:  <http://xxx.com:80/xyz.svc> |

## List of Web APIs

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Purpose** |
| 1 | GetMainTerminalData | Get data for main screen |
| 2 | GetAppointments | Get all appointments |
| 3 | GetAuctionList | Appointments with Preferred time |
| 4 | GetWaitList | Absentees |
| 5 | GetDirectList | Unreserved appointments created from Terminal |

## API Parameters

### Data types

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | |
| String | jSON string type | |
| Integer | jSON integer type | |
| Time | jSON string with the format hhmm in 24 hours format | |
| Date-Time | jSON date-time string with the format YYYYMMDDhhmm, time in 24 hours format. | |
| Language | jSON enumeration with values “en” | |
| Status | jSON integer with following possible values:  0 – wait-list  1 – Direct | |
| LogType | jSON integer with the following possible values:  1 – Information  2 – Error | |
| Status | jSON integer with the following possible values: | |
| 0 – Success | |
| 1 – Error | Display error screen |
| 2 – Fatal error   * Authentication failed * Terminal not mapped to any room | Display splash screen |

# User interface

User interface has to exactly match the current UI design of the hybrid version of the System in terms of UI elements, styles, captions, error messages, orientation and animations. This section is just a quantitative description of requirements. Please refer to the current screens for the exact quality requirements.

## Salient features

1. All unattended popup screens should automatically close after 3 minutes.
2. Command buttons of popup screens should be aligned exactly at the same XY position of main screen’s book/start button.
3. All close buttons should be aligned at the same XY position.
4. Control animations are carefully designed to capture and guide the attention of user. Current animations should be strictly followed.

## Status messages

Status messages are the messages displayed on screen.

**Note:**

Status messages are different from log messages. Log messages are for trouble shooting purpose and must be generated by combining an informative message plus the actual error returned.

# Appendices

## Status code categories

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Error screen** | **Splash screen** | **Log** |
| UI Validation | Yes | No | No |
| Connection | No | Yes | Yes |
| Communication | No | Yes | Yes |
| System | No | Yes | Yes |

## Status codes

Please refer to reference document #5 for HTTP status codes.

|  |  |  |
| --- | --- | --- |
| **Code** | **Scope** | **Description** |
| 101 | UI Validation | Past appointment validation |
| 102 | UI Validation | Booking time slot validation |
| 103 | UI Validation | Booker ID length validation |
| 104 | UI Validation | Mobile number validation |
| 105 | UI Validation | Email validation |
| 400 | Communication | HTTP Status 400 - Bad request |
| 401 | Communication | HTTP Status 401 - Unauthorized |
| 403 | Communication | HTTP Status 403 - Forbidden |
| 404 | Communication | HTTP Status 404 - Not found |
| 408 | Communication | HTTP Status 408 - Request timeout |
| 413 | Communication | HTTP Status 413 - Request entity too large |
| 500 | Communication | HTTP Status 500 - Internal server error |
| 502 | Communication | HTTP Status 502 - Bad gateway |
| 503 | Communication | HTTP Status 503 - Service unavailable |
| 599 | Communication | All other HTTP error codes |
| 600 | Connection | Wi-Fi is not available |
| 601 | Connection | Could not establish connection to API server (E.g. Invalid IP/Port) |
| 602 | Connection | Connection to API server timed out (E.g. Server busy or Invalid IP/Port) |
| 603 | Communication | Connection established, but server did not return any data, or no internet access, or some other error occurred in communication. |
| 610 | Connection & Communication | All other errors returned by OS system call |
| 999 | System | Rest of the errors returned by OS system call |

# References

[http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html" \l "sec10)

<http://www.jqwidgets.com/jquery-widgets-demo/demos/jqxscheduler/index.htm>