PES UNIVERSITY		**Document Ref.:**	SRS008
		Version No.:	v01
		Date:	15.10.2019
		Copy No.:	001
Project Name:	Hospital Information Management System		
Project Code:	008		
Status:	**Draft** / ~~Current~~ / ~~Superceded~~		
Document Type:	~~Controlled~~ / **Uncontrolled**		

Low Level Design Document for a Hospital Information Management System (HIMS), focussed at the Hospital Patient Information Management Module (HPIMS)

This document is aimed at documenting the Low Level Design corresponding to a Hospital Information Management System, primarily focussed at the Patient Information Management Module.

Prepared By: Team 008		**Reviewed By:**	
Name		**Name**	**Date**
Name	**Date**	**Dr. Ananth Koppar**	
A Abhishek	15.10.19		
A Om Sai Vivek Reddy	15.10.19	**Approved By:**	
Abhishek Narayanan	15.10.19	**Name**	**Date**
Abhishek Prasad	15.10.19	**Dr. Ananth Koppar**	
Abijna Rao	15.10.19		
Adapa Shivani	15.10.19		
Project Representative(s)		**PESU Representative(s)**	
1. A Abhishek 2. A Om Sai Vivek Reddy 3. Abhishek Narayanan 4. Abhishek Prasad 5. Abijna Rao 6. Adapa Shivani		**Dr. Ananth Koppar**	

TABLE OF CONTENTS

Definitions, Acronyms and Abbreviations

This section provides for definition of all terms, acronyms and abbreviations required for interpreting the Low Level Design Document. Well known abbreviations need not be stated.

This section provides for definition of all terms, acronyms and abbreviations required for interpreting the CRS. Well known abbreviations have not been stated.

1. **SMTP**- Simple Mail Transfer Protocol
2. **HIMS** - Hospital Information Management System
3. **HPIMS** - Hospital Patient Information Management System
4. **HTTPS**- Hyper Text Transfer protocol secure
5. **TLS**- Transport Layer Security
6. **OPD**- Outpatient Department
7. **IPD**- In Patient Department
8. **VPN**- Virtual Private Network
9. **SFTP**- Secure File Transfer Protocol
10. **NABH**- National Accreditation Board for Hospitals & Healthcare Providers
11. **JCI**-Joint Commission International
12. **HIPAA**- Health Insurance Portability and Accountability Act
13. **HL7**- Health Level 7

References

This section describes the complete list of documents referred to prepare the Low Level Design. The reference documents shall describe the title, version number, dates, authors and publishers, whatever is applicable. The Standards used for design shall also be clearly defined.

This section describes the complete list of documents referred to prepare the CRS.

- Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, Communications of the ACM (March), 338-348. Quoted: 03.03.1990
- Murat M. Tanik, Eric S. Chan, Fundamentals of Computing for Software Engineers, Van Nostrand Reinhold, 1991. ISBN: 0-442-00525-3

  URL: http://www.cs.helsinki.fi/u/przybils/courses/CBD06/papers/01184164.pdf

  (Quoted, 18.05.2012)
- Digital Library: http://computer.org/publications/dlib. (Quoted, 18.05.2012)
- Ridley, M. (2006). Requirement analysis and specification Guide (Crocus Information Limited,
- Devon, UK). URL: http://www.cilco.co.uk/index.html (Quoted, 18.05.2012)
- Spriestersbach, A. (2009) Component Design and Open Specification (Resource development and Management), Seventh Framework Program. FPF-ICT-2009-5. A research sponsored by the European Union.

  URL:http://4caast.morfeoproject.org/wpcontent/uploads/2011/09/4CaaSt_D4.2.1_Components_Design_and_Open_Specification.pdf. (Quoted, 18.05.2012)
- Royce, W (1970). Managing the development Of Large Software System. IEEE WESCON
- proceeding page 1-9
- Archer, P. (2009) Personalized Access to Cultural Heritage Space. Project sponsored by European Union. Grant agreement number: ICT-2009-270082
- Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf.(Seattle, WA, 1-5 April), 249-256. Quoted: 03.03.1990
- Boehm, B. (1989), Software Risk Management, IEEE Computer Society Press, Los Alamitos, CA.
- Goa, J. (2002) Software Integration and Testing, San Jose state University.URL:

  http://www.engr.sjsu.edu/gaojerry (Quoted, 18.05.2012)

➢ Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. Proc. ACM CHI'94

Conf.(Boston, MA, April 24-28), 152-158.

➢ Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY.

URL: http://www.useit.com/papers/heuristic/heuristic_list.html (Quoted, 18.05.2012)

➢ P. Graham, B. Nelson, and B. Hutchings, “Instrumenting bit streams for debugging FPGA

circuits,” in Proc. of IEEE Symposium. on Field-Programmable Custom Computing Machines

(FCCM), 2001 .

➢ Society for Risk Analysis (SRA) , 2002.

URL: https://acc.dau.mil/CommunityBrowser.aspx?id=17607 (Quoted, 18.05.2012)

➢ Mullins, C. 2002. Database Administration

Published by Pearson Education Corporate Sales Division, ISBN: 0-201-74129-6 Quoted: 15.12.2011

➢ Vieira, R. (2007). SQL Server 2005 Programming, Professional, ISBN:0-7645-8434-0

Quoted: 15.12.2007

➢ Molina, H., Ullman, J. & Widom, J. Database System The Complete Book 2nd edition

Published by Pearson Education International, ISBN (13): 0978-0-13-135428-9 Quoted: 02.03.2012

➢ Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90

Conf. (Seattle, WA, 1-5 April), 249-256. Quoted: 15.4.2012

➢ Connolly, T. & Begg, C. Database System 5th edition

Published by Pearson Education, ISBN (13): 978-0-321-52306-8 Quoted: 12.03.2012

➢ Elmasri, R. & Navathe, S. Database System 6 th edition Published by Berkeley, CA: Apress.

ISBN: 0-13-214498-0 Quoted: 15.4.2012

➢ Bersoff, E. H. (1984). "Elements of Software Configuration Management.” IEEE Trans. Software

Engineering, vol. SE-10, no. 1, January, pp. 79-87

➢ Kan, S. (2002) Metro and Model in Software Quality Engineering, Addison-Wesley Professional; 2

editions (September 30, 2002) ISBN-10: 0201729156 Quoted: 18.5.2012

➢ Ratcliffe, A. (2011) SAS Software Development with the V-Model, SAS Global Forum 2011.

➢ Microsoft Corporation (2012).

URL: http://msdn.microsoft.com/en-us/library/ee382825.aspx

Change History

This section describes the details of changes that have resulted in the current low-level Design document.

#	Date	Document Version No.	Change Description	Reason for Change
1.				
2.				
3.				

1.0 Introduction

1.1 Overview

This section provides a brief insight about the document as a whole, its purpose and the target audience relevant to this document. This document is aimed at documenting the High Level Design pertaining to a Hospital Information Management System, primarily focussed at the Patient Information Management Module.

Components of the healthcare industry which are primarily dependent on paper based documentation, primarily for patient information management are plagued by the following menacing problems :

1. **Lack of immediate retrieval means**: In order to retrieve data related to a patient's history, the hospital staff has to go through heaps of files and registers which is not only inconvenient but results in wastage of time.
2**. Lack of means for immediate information storage** : The information generated by various transactions takes time and effort to be stored at an appropriate place.
3. **Lack of prompt updation methods** : Various updations pertaining to patient details are tedious to make as paperwork is involved.
4. **Error prone manual calculation** : Manual calculations involved in billing, etc.are error prone and take a lot of time this may result in incorrect information.
5. **Preparation of accurate and prompt reports** : This becomes a difficult task as information is difficult to collect from heaps of registers which may have been stored irregularly in an unorganized manner.

Therefore, in order to tackle these issues, our objective is to develop a world class patient information management suite comparable to the state-of-the-art systems, driven by Internationally recognized standards with a seamlessly integrated architecture with the motive of providing a one stop paperless solution for patient information management meeting meaningful standards for reducing errors creeping into data entry and storage and preventing data loss, enhancing efficiency and productivity of the processes involved using optimal resources. Our software system is also focussed at improving ease of communication between patients and the hospital staff, increasing transparency and credibility employing state-of-art customizable, scalable and flexible technology with robust backup and storage features and the ease of use is targeted towards facilitation of execution of day to day routines in a convenient and hassle free manner. The primary target audience of the system are the patients, whose convenience is critical

to ensure customer satisfaction and a wide variety of hospital staff at various roles in the hierarchy, ranging from receptionists to doctors, nurses and administrators.

This document is to be read by the development team, the project managers, marketing staff, testers and documentation writers. Our stakeholders, company manufacturing associated hardware, company providing embedded operating system, shareholders, and distributors who markets the finished product, may review the document to learn about the project and to understand the high level design pertaining to the software.

1.2 Scope

This section describes all aspects of high-level design that has been undertaken.
The proposed software is the core part of a larger Hospital Management System as the scope of this project is limited and targeted primarily at Patient Information Management. However the proposed system can be conveniently integrated with the other participating components to for a complete HIMS.
The HPIMS is targeted towards two broad classes of patients namely, in-patients and out-patients. The system is meant to keep track of all essential details of both in-patients and out-patients including Patient id, name, date of birth, gender, city name, state name, phone number, id of doctor assigned, etc. to name a few, more of which have been elaborately portrayed in further sections. The system should facilitate convenient entry, storage and future retrieval of relevant information and thereby minimize the usage of paperwork in order to speed up patient handling and other daily routines at a low cost and employing minimal resources in an optimal way.
The primary features to be incorporated in such a system and the scope of this software include but are not limited to the addition, removal or updation of relevant patient information, appointment scheduling and management, elaborate dashboards for patients, portraying various essential details such as appointment, schedule, bed and ward allocation details in case of in-patients, medication etc. The system should also provide dashboards for the administrators and front desk staff so that they can keep track of the day to day proceedings of the organization and might even conveniently perform analytics to detect inefficiencies and optimize them.
In this document, we have vividly portrayed the high level design required to build such a software and have modelled the system using various UML modelling techniques including use case diagrams, class diagrams, sequence diagrams, Entity Relationship diagrams, not only pertaining to the system as a whole but for each module of the proposed system. The design considerations about the user interface and the external interface at modular level has also been provided.

2.0 Design Constraints, Assumptions and Dependencies

This section provides the list of constraints, assumptions and dependencies.

This section of the CRS shall provide a general description of any other item that will limit the developer's option for designing the system.

This section shall list each of the factors that affect the stated requirements. These factors are not design constraints on the software but changes to them can affect the requirements.

General Constraints and Security consideration:-

- The system must be delivered by the deadline.
- Only the administrator should be able to configure the software.
- The administrator should provide read and write access to all the staff
- There will be a 2 way authentication for the administrator - a login dashboard having a username and password field along with an OTP(one time password) generation sent to the registered email id of the administrator.
- The system must comply with Joint Commission International (JCI) standards define the performance expectations, structures, and functions that must be in place.
- The system must comply with NABH standards which provide a framework for quality assurance and quality improvement for hospitals. The standards focus on patient safety and quality of care. The standards call for continuous monitoring of sentinel events and comprehensive corrective action plan leading to building of quality culture at all levels and across all the functions.
- The system must comply with HIPAA, which is a series of regulatory standards that outline the lawful use and disclosure of protected health information (PHI). HIPAA compliance is regulated by the Department of Health and Human Services (HHS) and enforced by the Office for Civil Rights (OCR)

System Constraints :

The system must comply with at least the following requirement specifications :

CPU - Quad core (8 core)
Memory - 16GB
Hard disk - 500GB
Cache memory -
L1 D/I cache - 32KB per core
L2 cache - 256KB per core
L3 Cache - 16MB (could be shared across all cores or may)

Battery - APC Smart-UPS SMT1500
LAN Ethernet cables- Cat 6a

Assumptions and Dependencies :-

- It is assumed that compatible computers will be available before the system is installed and tested.
- It is assumed that the Hospital will have enough trained staff to take care of the system

3.0 Design Description

This section describes the design with respect to functional modules.

3.1 Module : In Patient Management :-

In-patient module provides the ability to admit, discharge and transfer a patient. The user of the module can assign ward and bed to the selected patient. The user can also view the summary information of the patient along with his/her vitals, diagnosis.

This module will be used by the nurses or other clinicians to monitor the in-patients.

The primary benefits of this feature are-

- View patient list based on different status - To Admit, Admitted, To Discharge etc.
- Admit a patient and assign a bed
- Discharge a patient
- Transfer a patient from one bed to another
- View a summary of patient information

3.1.1 Class : Nurse

3.1.1.1 Class Description

Nurses are responsible for assigning patients to appropriate wards if the beds are available, otherwise putting patients on the waiting list. Nurses can also discharge a patient. They can also reallocate a patient to a different ward or bed.

...

3.1.1.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	name	private	NULL	Contains the name of the nurse.
String	nurseId	private	NOT NULL	Contains the nurseId of the nurse.

3.1.1.2.1 Method : Admit Patient

The following details shall be defined for the methods:

- Purpose
- Input
- Output
- Parameters
- Exceptions
- Pseudo-code

Purpose :-

This method will help the user to admit the patient.

Input :-

Input will be patient registration Id.

Output :-

The given patient will now show up in the “Admitted” section and will be removed from the “To Admit” section.

Parameters :-

Patient registration Id.

Exceptions :-

If there is no free bed present in the ward the patient will continue to be present in the “To admit” section and the operation “To Admit” stands cancelled.

3.1.1.2.2 Method : Discharge Patient

Purpose :-

This method will help the user to discharge the patient.

Input :-

Input will be patient registration Id.

Output :-

The given patient will be removed from the “Admitted” section.

Parameters :-

Patient registration Id.

Exceptions :-

If the patient Registration Id is not found under the “admitted’ section, the operation stands cancelled. No changes will be made.

3.1.1.2.3 Method : Transfer Patient

Purpose :-

This method will help the user to transfer the patient from one bed to another.

Input :-

Input will be patient registration Id , ward name , bed number.

Output :-

The given patient bed no. as well as ward name will be updated. The previous bed will be updated as free.

Parameters :-

Patient registration Id , Ward Id , bed number.

Exceptions :-

If the user while changing the bed finds that no bed is free then the operation stands cancelled and the user would not be allowed to transfer the patient.

3.1.2 Class : Ward

3.1.2.1 Class Description

This class contains the details of the ward present in the hospital i.e , Name , Ward Id , Location. It helps the user by displaying the details of the ward i.e , the no.of bed present in the ward , the no.of free bed as well as allocate and deallocate beds in the ward.

3.1.2.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	Id	private	NOT NULL	Contains the name of the Ward.
String	Name	private	NULL	Contains the Ward Id of the ward.
String	Location	private	NULL	Contains the location of the ward

...

3.1.2.2.1 Method : Get_no_of_free_beds

Purpose :-

This method will help the user to see the number of free beds present in a ward.

Input :-
Ward Id

Output :-
The no.of free beds will be presented to the user along with their bed Number..

Parameters :-
Ward Id

Exceptions :-
If the Ward Id is not present an error is displayed saying “Ward not Present”.

3.1.2.2.2 Method : Allocate_free_bed

Purpose :-
This method will help the user to allocate a free bed, if present in the ward.

Input :-
Bed No , PatientId

Output :-
The Bed status will be changed to “allocated” along with the PatientId.

Parameters :-
Bed No , PatientId

Exceptions :-

If the user while allocating a bed finds that no bed is free then the operation stands cancelled and the user would not be allowed to allocate the bed.

3.1.2.2.3 Method : Deallocate_free_bed

Purpose :-
This method will help the user to deallocate an occupied bed.

Input :-
Bed No

Output :-
The Bed status will be changed to “free”.

Parameters :-
Bed No

Exceptions :-

If the user while deallocating a bed finds that the bed is already free, then the operation stands cancelled.

3.1.3 Class : Bed

3.1.3.1 Class Description

This class contains the information about Bed i.e , the bed number and the ward name in which it is present. Also it is used to display the information details of the patient who is allocated that bed.

3.1.3.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
Integer	BedNo	private	NOT NULL	Contains the Bed number of the bed.
String	WardId	private	NULL	Contains the Ward Id of the ward.

...

3.1.3.2.1 Method : Allocate_patient

Purpose :-
This method will help the user to allocate a free bed to a patient.

Input :-
Bed No , PatientId

Output :-

The Bed status will be changed to “allocated” along with the PatientId.

Parameters :-
Bed No , PatientId

Exceptions :-

If the user while allocating a bed finds that no bed is free then the operation stands cancelled and the user would not be allowed to allocate the bed.

3.1.3.2.2 Methods : Deallocate_patient

Purpose :-
This method will help the user to deallocate an occupied bed.

Input :-
Bed No

Output :-
The Bed status will be changed to “free”.

Parameters :-
Bed No

Exceptions :-

If the user while deallocating a bed finds that the bed is already free, then the operation stands cancelled.

3.1.3.2.3 Methods Get_Bed_Info

Purpose :-
This method will help the user to see the details of the patient admitted on that bed.

Input :-
Bed No

Output :-
The method will display the details of the patient(Patient Dashboard).

Parameters :-
Bed No

Exceptions :-

No dashboard will be displayed in case the bed is unoccupied.

3.1.4 Class : Patient

3.1.4.1 Class Description

This class will help the user to view the patient details.

3.1.4.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
Integer	Id	private	NOT NULL	Contains the Registration Id of the patient.
String	Name	private	NULL	Contains the name of the Patient.
Integer	Age	private	NULL	Contains the age of the Patient.
String	Address	private	NULL	Contains the address of the patient.
String	BloodGroup	private	NULL	Contains the Blood group of the patient.
String	Doctor	private	NULL	Contains the ID of the doctor.
Integer	Mobile No	private	NULL	Contains the Mobile number of the patient.

3.1.4.2.1 Methods : Display_Patient_Info

Purpose :-
This method will help the user to see the details of the patient.

Input :-
Patient Registration ID

Output :-
The method will display the details of the patient(Patient Dashboard).

Parameters :-
Patient Registration ID

Exceptions :-

No dashboard will be displayed in case no patient with the given Registration Id is found.

3.2 Module : Registration

3.2.1 Class : user

3.2.1.1 Class Description

User class is a general class which has child classes which use attributes present in the user class and can have additional attributes if required. This class describes the user with certain attributes like username, userId, gender, DOB etc.

3.2.1.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	username	private	NULL	Contains the name of the user..
String	userId	private	NOT NULL	Contains the userId of the user .
String	password	private	NULL	contains the password of the user
String	personId	private	NULL	contains the unique id of the person operating system.
String	FirstName	private	NULL	contains the first name of the user
String	LastName	private	NULL	contains the last name of the user
String	IDcard	private	NULL	contains the ID of the card given to the user
String	Gender	private	NULL	contains the gender of the user
String	DOB	private	NULL	Contains the date of birth of the user
String	contact	private	NULL	contains the contact of the user.

3.2.1.2.1 Method : Login

The following details shall be defined for the methods:

- Purpose
- Input
- Output
- Parameters
- Exceptions
- Pseudo-code

Purpose :-

This method will allow the user to login to the system..

Input :-

Input will be userId and password.

Output :-

The user will be logged into his/her account with the given credentials.

Parameters :-

user Id and password.

Exceptions :-

If the userId or password will be wrong then the user will not be logged into his/ her account successfully..

3.2.1.2.2 Method : Select_OPD_visit

Purpose :-

This method allows the user to select the OPD visit from the case approached the hospital..

Input :-

Choose the option from the three types of OPD visits like emergency, consultant, normal visit.

Output :-

The given option is taken and the patient is provided with a chosen type of visit..

Parameters :-

Type of visit..

Exceptions :-

If the type of visit is not under the three given options, then the user should explicitly write it in the written section..

3.2.1.2.3 Method : Upload Photograph

Purpose :-

This method will allow the user to upload photograph of the patient.

Input :-

Input will be the photograph of the patient to be uploaded.

Output :-

The uploaded photo is shown on the screen of the user.

Parameters :-

Patient photograph which is taken from the webcam.

Exceptions :-

If the visit is of emergency type then the patient photo might not available for uploading

3.2.1.2.4 Method : generateQRcode

Purpose :-

This method will generate a unique QR code for each patient visited the hospital..

Input :-

Input will be patient details and the request for QR code generation.

Input to this module would be the patient registration Id and all the registered details about the patient including the following :

- Name of the patient broken into ”First Name” , “Middle Name” and “Last Name”.
- Gender as male or female or transgender.
- The Date of Birth in Day-Month-Year format
- Age in years
- Phone number
- Emergency Contact number
- Patient's Address broken down into 5 fields - House number,Line number , Pincode , City, State. Smoking history checkbox
- Father’s/Husband name
- An identification mark
- Any additional note for the patient.

Output :-

The QR code generated is shown on the screen.

Parameters :-

Firstname, last name, gender, date of birth age, phone number, emergency contact, address, father/husband name, identification mark.

Exceptions :-

If any of the necessary patient details are not given then the QR code generation will not happen..

3.2.1.2.5 Method : Print_Patient_Card

Purpose :-

This method will print the patient card will all the patient details and the type of visit, cost of the visit, QR code, patientID..

Input :-

Input will be the details of the patient like firstname, last name, date of birth, gender, address, age.
.

Output :-

The patient will be given a printed card with details of the patient and related to the visit to the doctor.

Parameters :-

Patient details like Firstname, last name, gender, date of birth age, phone number, emergency contact, address, father/husband name, identification mark .

Exceptions :-

If all the necessary patient details are not given to the user then the card will not be printed.

3.2.2 Patient

3.2.2.1 Class Description

This class consists of the details of the patient who is joining in the hospital. This class consists of attributes like patientid, weight, height, blood type, disease.

3.2.2.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	patientID	private	NOT NULL	Contains the patientID of the patient

Int	height	private	NULL	Contains the height of the patient.
Int	weight	private	NULL	contains the weight of the patient.
String	blood type	private	NULL	blood type of the patient.
String	disease	private	NULL	contains the disease or problem of the patient.

...

3.2.2.2.1 Method : choose_the_doctor

Purpose :-

This method will let the patient give the preference of the doctor from past experience of the patient.

Input :- The input will be the doctor names of the patient preference upto the count of three.

Output :-

The patient will be given the appointment with the prefered doctor if the doctor is currently available.

Parameters :-

The method is given with the prefered doctor names chosen by the patient..

Exceptions :-

If the preferred doctor is currently unavailable then the patient is given an appointment with another doctor.

...

3.2.2.2.2 Methods : Make_appointment

Purpose :-

This method will let the patient book an appointment before the visit to the doctor according to available visiting hours.

Input :- The input will be a request for the appointment with the doctor..

Output :-

The patient will be given with the confirmation of the appointment or message of failure of the appointment booking..

Parameters :-

Time at which appointment is required..

Exceptions :-

If the time slot at which the appointment requested is not available then the appointment is not provided..

3.2.2.2.3 Methods : cancel_appointment

Purpose :-

This method will let the patient give the preference of the doctor from past experience of the patient.

Input :- The input will be the doctor names of the patient preference upto the count of three.

Output :-

The patient will be given the appointment with the prefered doctor if the doctor is currently available.

Parameters :-

The method is given with the prefered doctor names chosen by the patient..

Exceptions :-

If the preferred doctor is currently unavailable then the patient is given an appointment with another doctor.

3.2.2.2.4 Methods : change_appointment

Purpose :-

This method will let the patient change the appointment timing after the previous booking if the patient is willing to change.

Input :- The input will be the changed time for appointment.

Output :-

The patient will be given an appointment with the new changed time if the time slot is available.

Parameters :-

The method is given with the present time and changed time of appointment.

Exceptions :-

If the changed time slot for appointment is not available then the patient is denied for the request of the change and the previous time is fixed.

3.2.2.2.5 Methods : Register

Purpose :-

This method will let the patient to register himself with the hospital website if he/ she is/ are new users and would like to have an account in the hospital website.

Input :- The input will be the necessary details as per the hospital website requirement like first name, last name, age, gender, problem, phone number, address.

Output :-

The patient will be created and given a new account with a unique patientID attached to the account.

Parameters :-

The method is given the details like first name, last name, phone number, age, gender, address, problem.

Exceptions :-

If the patient does not provide some necessary which are must for the creation of the account for the patient, then the account will be created.

3.2.3 Class: FrontDeskUser

3.2.3.1 Class Description

This class list the attributes which describe the FrontDeskUser like userId, license no. deptId, workdate, workshift, position, username, password. This class has methods which help the front desk user to schedule patient appointments, change patient appointment, registering the patient, takeDetails.

3.2.3.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	username	private	NULL	Contains the name of the user..
String	userId	private	NOT NULL	Contains the userId of the user .
String	password	private	NULL	contains the password of the user
String	Licence no.	private	NULL	contains the unique id of the person operating system.
String	FirstName	private	NULL	contains the first name of the user

Int	DeptID	private	NULL	contains the last name of the user
Date	Workdate	private	NULL	contains the ID of the card given to the user
String	workshift	private	NULL	contains the gender of the user
String	Position	private	NULL	Contains the date of birth of the user

3.2.3.2.1 Methods: addSchedule

Purpose :-

This method will let the frontdesk user to add schedule for the doctor availability and let the patient take appointment accordingly .

Input :- The input will be the time slot for each doctor and their availability.

Output :-

The output will be the schedule available for the patient to look after and make an appointment accordingly.

Parameters :-

The method is given with the doctorid and the corresponding timesolt of availability of doctor.

Exceptions :-

If there is a particular doctor not available on a particular day then his schedule is not added.

3.2.3.2.2 Methods: changeSchedule

Purpose :-

This method will let the frontdesk user to change the existing schedule and add a new schedule.

Input :- The input will be the new schedule which should be replaced by the existing one.

Output :-

The output will be the change schedule for the patients to check in.

Parameters :-

The method is given with new schedule according to the doctor availability.

Exceptions :-

If there is no change in doctor availability then there is no need for change in schedule.

3.2.3.2.3 Methods: RegisterPatient

Purpose :-

This method will let the user to register the patient who is new to the hospital.

Input :- The input will be the details which are taken from the patient like firstname, last name, age, gender, address, phone no. problem.

Output :-

The patient is created and given a unique patient ID by the user.

Parameters :-

The method is given with the parameters like first name, last name, phone no. age, gender, blood type, address.

Exceptions :-

If the patient is already a registered one he/ she will not be newly registered.

3.2.4 Class: Administrator

3.2.4.1 Class Description

This class has the attributes adminId, username, password. This class has methods queryPatient list used to fetch the patient list, assignDoctor is used to assign a doctor to the patient, generate fee is used to provide total cost, register patient method is used to register the patient.

3.2.4.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	AdminID	private	NOT NULL	Contains the AdmintID of the admin
String	username	private	NULL	Contains the username of admin
String	password	private	NULL	contains the password for the admin account.

3.2.4.2.1 Methods: QueryPatientList

Purpose :-

This method will let the admin query the patient list to find out if the patient is already registered or not.

Input :- The input will be patientID or patient username.

Output :-

The output will be yes/ no and patient details depending upon if the patient is already registered or not.

Parameters :-

The method is is input with patientID or patient name..

Exceptions :-

If the patient is not registered before then the method does not retrieve the patient list.

3.2.4.2.2 Methods: assignDoctor

Purpose :-

This method will let the admin assign a doctor to the patient depending on the patient preference of the doctor.

Input :- The input will be doctorID or doctor name.

Output :-

The output will be yes/ no and patient details depending upon if the patient is already registered or not.

Parameters :-

The method is input with the doctorId or doctor name as the parameter..

Exceptions :-

If the doctor whom the patient has asked for is not available then he/ she will not be assigned with the respected doctor.

3.2.4.2.3 Methods: generateFee

Purpose :-

This method will let the admin generate the fee that is to be paid by the patient.

Input :- The input will patientID, patient details regarding the visit to hospital.

Output :-

The output will be total amount of fee the patient should pay to the hospital.

Parameters :-

The method is input with the patientID from which the system will fetch the medical history of the patient and generate the fee accordingly.

Exceptions :-

If the doctor the patient has already paid the bill then he/ she need not check again.

3.2.5 Class: Registration

3.2.5.1 Class Description

Registration class has necessary attributes which need to be submitted by the patient for registration process.

3.2.5.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	Firstname	private	NOT NULL	Contains the Firstnam of the patient
String	Last name	private	NULL	Contains the Last name of patient
String	password	private	NULL	contains the password of the patient account.
String	username	private	NULL	Contains the username of the patient
String	BloodType	private	NULL	Contains the BloodType of the patient
String	phone no.	private	NULL	Contains the phone no. of the patient
String	gender	private	NULL	Contains the gender of the patient
String	address	private	NULL	Contains the address of the patient

3.3 Module : Patient Listing

This module enables the user to view the list of patients registered at the facility. The module must be equipped with filtering capabilities. The system must enable the user to navigate to the Patient Dashboard to view and add clinical information for the patient by selecting the corresponding patient info as displayed in the patient list.

Patient Lists must provide the following functionalities :

1. A snapshot of patients belonging to various categories, must be rendered by the system either as a tile or list view based on the user's preferences..
2. The filter to be applied in order to display the patient list must be configurable. The variety of filters provided by the functionality include :
    - list of all patients who have an active visit
    - all patients who have been sent to a particular department
    - list of in-patients
    - list of out-patients
    - list of patients who have visited the facility for consultation between a certain range of dates
    - list of patients who have been admitted between a certain date range
    - list of patients who have been discharged within a certain timeframe

3.3.1 Class : User

3.3.1.1 Class Description

This class represents the user of the system who needs the functionality of viewing list of patients who are admitted/discharged in a certain date range or searching for a patient's details. The user may broadly be classified specifically into two classes, the administrator who might need to view patient lists and might possibly perform relevant analytics and secondly the front-end staff who might query the patient list for some patient's details.

3.3.1.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	name	private	NULL	name of the user
String	employeeTYpe	private	NULL	type of employee : admin r front-desk staff
String	designation	private	NULL	designation of employee
String	employeeSSN	private	NULL	SSN of user
String	email	private	NULL	email id of user
String	phoneNumber	private	NULL	contact number of user

3.3.1.2.1 Method : createDashboard

The following details shall be defined for the methods:

- Purpose
- Input
- Output
- Parameters

- Exceptions
- Pseudo-code

Purpose :-

This method is entitled to be invoked whenever a new entry of a patient is made as registration, such that a unique dashboard for the patent is set up for future reference by the patient, doctor or concerned staff .

Input :-

Input to this module would be the patient registration Id and all the registered details about the patient including the following :

- Name of the patient broken into ”First Name” , “Middle Name” and “Last Name”.
- Gender as male or female or transgender.
- The Date of Birth in Day-Month-Year format
- Age in years
- Phone number
- Emergency Contact number
- Patient's Address broken down into 5 fields - House number,Line number , Pincode , City, State. Smoking history checkbox
- Father’s/Husband name
- An identification mark
- Any additional note for the patient.

Output :-

The dashboard is not displayed at this stage and is invoked only when the viewDashboard method is called. This method is just in-charge of creation of the skeleton dashboard which would be subsequently updated as the patient visits or consults doctors, undergoes lab tests or is admitted and subject to medical services.

Parameters :-

Input to this module would be the patient registration Id and all the registered details about the patient including the following :

- Name of the patient broken into ”First Name” , “Middle Name” and “Last Name”.
- Gender as male or female or transgender.
- The Date of Birth in Day-Month-Year format
- Age in years
- Phone number
- Emergency Contact number
- Patient's Address broken down into 5 fields - House number,Line number , Pincode , City, State. Smoking history checkbox
- Father’s/Husband name
- An identification mark
- Any additional note for the patient.
- An optional picture of the patient

Exceptions :-

Should indicate appropriate messages when dashboard creation is unsuccessful due to absence of required fields.

3.3.1.2.2 Method : viewDashboard

Purpose :-

This method is invoked when a user such as doctor or nurse wishes to view a patient's dashboard to review patient data, medical history and lab results.

Input :-

Input will be patient registration Id.

Output :-

The method displays the patient dashboard corresponding to the i=entered id.

Parameters :-

Patient registration Id.

Exceptions :-

If the patient Registration Id is not found when queried in the patient list, a relevant error message should be displayed.

3.3.1.2.3 Method : configureDashboard

Purpose :-

This method will help the user to update details to a patient's dashboard, such as upload test results, capture consultation data, etc.

Input :-

Input will be patient registration Id

Output :-

No output is displayed as such, the dashboard is internally updates in the database

Parameters :-

Patient registration Id

Exceptions :-

If the patient Registration Id is not found when queried in the patient list, a relevant error message should be displayed.

3.3.1.2.4 Method : filterBy

Purpose :-

This method will help the user to filter patients based on a certain filtering criteria fro the overall patient list

Input :-

Input will be filtering attribute/criteria

Output :-

The list of patients satisfying the filtering criteria or having a field same as the filtering attribute

Parameters :-

filtering attribute/criteria

Exceptions :-

If there are no patients found by a particular filter search, an appropriate error message should be displayed.

3.3.2 *Class : InPatientFilter*

3.3.2.1 Class Description

This class enables the user of the system who is needs the functionality of viewing list of in-patients who are admitted/ in the hospital.

3.3.2.1.1 Method : getInPatients

Purpose :-

This method will help the user to retrieve the list of inpatients admitted to the hospital

Input :-

None

Output :-

List of in-patient names along with their registration ids

Parameters :-

None

Exceptions :-

If there are no in-patients admitted at a certain point of time, the system should respond with an error message.

3.3.3 *Class : OutPatientFilter*

3.3.3.1 Class Description

This class enables the user of the system to view the list of out patients who visit the hospital for consultation or only for undergoing lab tests.

3.3.3.1.1 Method : getOutPatients

Purpose :-

This method will help the user to retrieve the list of out patientsvisiting the hospital

Input :-

None

Output :-

List of out-patient names along with their registration ids

Parameters :-

None

Exceptions :-

None

3.3.4 *Class : VisitDateFilter*

3.3.4.1 Class Description

This class enables the user to view the list of patients who visited the hospital on a particular date

3.3.4.1.1 Method : getVisitorsOn

Purpose :-

This method will help the user to retrieve the list of patients visiting the hospital on a certain specified date.

Input :-

Date

Output :-

List of patient names along with their registration ids who visited the hospital on the specified date

Parameters :-

Date

Exceptions :-

If there are no patients visiting, the system should respond with an error message.

3.3.5 *Class : DischargeDateFilter*

3.3.5.1 Class Description

This class enables the user of the system to view the list of patients who got discharged between a certain range of dates.

3.3.5.1.1 Method : getischargesBetween

Purpose :-

This method will help the user to retrieve the list of patients discharged from the hospital between a given start date and an end date

Input :-

start date and end date

Output :-

List of in-patient names along with their registration ids who got discharged between the specified time interval

Parameters :-

start date and end date

Exceptions :-

If there are no in-patients discharged at a certain point of time, the system should respond with an error message.

3.3.6 *Class : AdmissionDateFilter*

3.3.6.1 Class Description

This class enables the user of the system to view the list of patients who got admitted to the hospital between a specified range of dates.

3.3.6.1.1 Method : getAdmitsBetween

Purpose :-

This method will help the user to retrieve the list of patients admitted to the hospital between the given dates

Input :-

start date and end date

Output :-

List of in-patient names along with their registration ids

Parameters :-

start date and end date

Exceptions :-

If there are no in-patients admitted at a certain point of time, the system should respond with an error message.

3.4 Module : Patient Billing

Doctors may prescribe medicines or drugs for patients or may additionally place orders for patients to undergo prescribed medical tests for patients using the consultation feature of the clinical module which is beyond the scope of this software. The patient would then proceed to the Pharmacy or to a Billing Counter in order to make the payment. This module describes the usage of the Billing and Accounting feature.

The following features must be supported by this module :

- View Sale Orders
- Add Discounts or Pay in Instalments
- Confirm Order

Once an order for a pharmacy product or medical test is confirmed, the system should not allow further editing of order information.

The module must follow the following workflow for printing bills/prescriptions of patients :
The system screen should display the patient in the list for whom medications or medical test orders were placed
The user would then select the patient from the patient list in order to view the invoice.
The subsequent screen should then allow the user to select the “Print Invoice” option in order to print the bill for the customer.

3.4.1 Class : User

3.4.1.1 Class Description

This class represents the user of the system who can view the sale orders of a patient, add discounts and confirm order before initiating payment.

3.4.1.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	name	private	NULL	name of the user
String	employeeTYpe	private	NULL	type of employee : admin or front-desk staff
String	designation	private	NULL	designation of employee
String	employeeSSN	private	NULL	SSN of user
String	email	private	NULL	email id of user
String	phoneNumber	private	NULL	contact number of user

3.4.2 *Class : Receptionist*

3.4.2.1 Class Description

This class enables the user of the system who needs the functionality of viewing the sale orders a patient is associated with, add discounts and confirm order before initiating payment.

3.4.2.1.1 Method : generateBill

Purpose :-

This method will help the user to generate an invoice for the patient portraying the medical services the patient has been subject to, the cost incurred, discounts added, total payable amount.

Input :-

patient registration id

Output :-

Invoice for the patient containing total payable amount and details of all of the services used by the patient.

Parameters :-

Patient registration id

Exceptions :-

If there are no medical services associated with the supplied patient id or if the patien id is non existent, an error message should be displayed.

3.4.2.1.2 Method : receivePayment

Purpose :-

This method will help the user to account for the payment made by the patient.

Input :-

payable amount

Output :-

Status of all due payable services should change to paid

Parameters :-

payable amount

Exceptions :-

None.

3.4.3 *Class : Bill*

3.4.3.1 Class Description

This class enables the system to keep track of the billing and accounting details of each patient

3.4.3.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	bill_no	private	NULL	unique invoice number
String	patientId	private	NOT NULL	unique id assigned to patient during registration
Float	labCharge	private	NULL	charges incurred due to lab tests
Float	nursingCharge	private	NULL	charges incurred due to nursing
Float	doctorCharge	private	NULL	charges incurred in doctor consultation
Float	totalAmount	private	NULL	total payable amount

3.4.4 *Class : Patient*

3.4.4.1 Class Description

This class enables the user to view the details of patients

3.4.4.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	name	private	NULL	name of the patient
String	Id	private	NOT NULL	unique id assigned to patient during registration
String	age	private	NULL	age of the patient
String	bloodgroup	private	NULL	blood group of the patient
String	mobileno	private	NULL	mobile number of the patient
String	email	private	NULL	email id of the patient
String	address	private	NULL	address of the patient

`

3.5 Outpatient Management :-

In outpatient management the user of the system can capture out patient's consultation data, schedule, create, or delete appointments with doctors, create patient visit dashboards, etc. The primary functionalities provided to the user are:

1. Enter Patient data retrospectively or on behalf of another provider.
2. View Patient Dashboard including Patient Details, Active and Past Programs.
3. View the summary of a Patient's visit along with graphs and trend mapping.
4. Capture specific clinical observations for the patient such as Obstetrics, Gynecology, etc.
5. Autocompletes available data fields for easier data entry.
6. Capture various diagnoses for the patient.
7. Capture Consultation notes for the patient.
8. Prescribe treatment orders for medications.
9. Place orders for radiology tests via PACS (Picture Archiving and Communication System) integration.
10. Place orders for Laboratory tests.
11. View the consultation history of the patient.
12. View scanned documents and results of tests.
13. Capture Bacteriology test results for the patient (different from laboratory tests). Bacteriology includes smear test results, culture test results and drug-sensitivity test results.
14. View Patient Lists with support for sorting by filters.

3.5.1 Class : Person

3.5.1.1 Class Description

It is a generalized class from which other subclasses such as Patient, Doctor, Receptionist and Administrator are inherited.

...

3.5.1.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	name	private	NULL	Contains the name of the person.
String	gender	private	NULL	Contains the gender of the person .
Date	date of birth	private	NULL	Contains the date of birth of the person .
String	phone	private	NULL	Contains the phone no of person.
String	address	private	NULL	Contains the address of the person .
String	email	private	NULL	Contains the email id of the person .

3.5.1.2.1 Method : Calculate Age

The following details shall be defined for the methods:

- Purpose
- Input
- Output
- Parameters
- Exceptions
- Pseudo-code

Purpose :-

In therapeutic purposes age plays a very vital role in determining the severity of disease. Hence HIMS has a method named calcAge which shall determine the age of a patient w.r.t his date of birth.

Input :-

Input will be patient registration Id, Date of Birth of the patient.

Output :-

Current Age of the patient.

Parameters :-

Input will be patient registration Id, Date of Birth of the patient.

Exceptions :-

If the user enters the date of birth wrongly (i.e greater than the current date) then the system will give an error.

3.5.2 Class : Patient

3.5.2.1 Class Description

It is a derived class from Person class. It is used to represent the patient. It consists of the attributes and methods of base class. Additionally it has attributes like history(i.e history of patient visits) and bloodGrp of the patient.

3.5.2.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
Integer	patient id	private	NOT NULL	Contains the unique id given to patient during registration.
String	history	private	NULL	Contains the history of patient visits to hospital Moreover it contains billing, consultation information during each of these visits.
String	blood group	private	NULL	Contains the blood group of the patient.

...

3.5.2.2.1 Method : display_patient_info

Purpose :-

It is used to display the patient information to doctors or receptionists.

Input :-

Patient ID

Output :-

Displays all the patient information

Parameters :-

Patient ID

Exceptions :-

If the user data isn't there in the database then an error message is displayed and end user is requested to enter the patient data before patient information gets displayed

3.5.2.2.2 Method : display_patient_history

Purpose :-

This method will help the doctor in diagnosing the patient better based upon the patient's previous visits.

Input :-

Patient Id

Output :-

The history of patient's previous visits along with specific details of each Patient ID is displayed.

Parameters :-

Patient Id

Exceptions :-

If the user data isn't there in the database then an error message is displayed and end user is requested to enter the patient data before patient history is displayed.

3.5.3 Class : Receptionist

3.5.3.1 Class Description

It is a derived class from Person class. It is used to represent the Receptionist working for the hospital. It consists of all the attributes and methods of base class. Additionally it has an attribute called employeeId which can be used by the employee to access resources of HIMS based on his/her privileges.

3.5.3.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
Integer	employeeId	private	NULL	Contains the employee Id of the receptionist.

...

3.5.3.2.1 Method : register_patient

Purpose :-

It is used by the receptionist to register new patients.

Input :-

Receptionist should enter required patient details(i.e all the attributes of Patient class).

Output :-

Success message incase the data gets committed into the database else retry message is displayed.

Parameters :-

All the required patient details(i.e all the attributes of Patient class).

Exceptions :-

Retry message is displayed in case the patient data is not added in to the database.

3.5.3.2.2 Methods : patient_status

Purpose :-

It is used by the receptionist to control the change in the appointment status to scheduled, CheckedIn, Missed, Completed or Cancelled.

Input :-

Patient ID

Output :-

Displays the patient's status .

Parameters :-

Patient ID

Exceptions :-

If the user data isn't there in the database then an error message is displayed and end user is requested to enter the patient data before patient status is displayed.

3.5.3.2.3 Method : schedule_Appintment

Purpose :-

It is used by the receptionist to schedule patient's appointments and allows them to access the requisite heath-care and medical services. Additionally it shall manage schedules of various health care service providers and for the medical services offered by them.

Input :-

Patient Id, service requested, preferable doctor.

Output :-

Success message is displayed incase the appointment is scheduled successfully.

Parameters :-

Patient Id, service requested, preferable doctor.

Exceptions :-

Retry message is displayed in case the appointment couldn't be scheduled.

3.5.3.2.4 Method : recieve_Payment

Purpose :-

It is used by the receptionist to store in the database stating that all medical dues are cleared by the patient.

Input :-

Patient ID, Amount paid, Mode of payment.

Output :-

Success message is displayed in case the database is updated successfully.

Parameters :-

Patient ID, Amount paid, Mode of payment.

Exceptions :-

Retry message is displayed in case the database wasn't updated.

3.5.3.2.5 Methods : display_dashboard

Purpose :-
It is used by the receptionist to view appointments schedules of a doctor.

Input :-
Employee ID of doctor.

Output :-
Displays the appointments scheduled corresponding to a doctor .

Parameters :-
Employee ID of doctor.

Exceptions :-
If the user data isn't there in the database then an error message is displayed and end user is requested to enter the doctor's data before doctor's appointments are displayed.

3.5.3.2.6 Method : cancel_Appointment

Purpose :-
It is used by the receptionist to cancel patient's appointments.

Input :-
Patient Id, Employee ID(doctor).

Output :-
Success message is displayed incase the appointment is cancelled successfully.

Parameters :-
Patient Id, Employee ID(doctor).

Exceptions :-
Retry message is displayed in case the appointment hasn't been cancelled.

3.5.3.2.7 Method : missed_Appointment

Purpose :-
It is used by the receptionist to change the status of patient's appointment to missed.

Input :-
Patient Id, Employee ID(doctor).

Output :-
Success message is displayed incase database is updated successfully.

Parameters :-
Patient Id, Employee ID(doctor).

Exceptions :-
Retry message is displayed in case the database hasn't been updated successfully.

3.5.3.2.8 Method : patient_CheckIn

Purpose :-
It is used by the receptionist to change the status of patient's appointment to checkedIn.

Input :-
Patient Id, Employee ID(doctor).

Output :-
Success message is displayed incase database is updated successfully.

Parameters :-
Patient Id, Employee ID(doctor).

Exceptions :-
Retry message is displayed in case the database hasn't been updated successfully.

3.5.3.2.9 Method : completed_Appointment

Purpose :-
It is used by the receptionist to change the status of patient's appointment to completed.

Input :-
Patient Id, Employee ID(doctor).

Output :-

Success message is displayed incase database is updated successfully.

Parameters :-
Patient Id, Employee ID(doctor).

Exceptions :-
Retry message is displayed in case the database hasn't been updated successfully.

3.5.4 Class : Doctor

3.5.4.1 Class Description

This class will help the user to view the doctor's details.

3.5.4.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
Integer	employeeId	private	NOT NULL	Contains the Employee Id of the doctor.
String	specialization	private	NULL	Contains the specialization of the Doctor.
Integer	yearsOfExp	private	NULL	Contains the number of years of experience of doctor.
String	qualification	private	NULL	Contains the qualifications of the doctor.
String	department	private	NULL	Contains the department of the doctor.

3.5.4.2.1 Methods : Display_contact_no

Purpose :-
This method will display the contact number of doctor which may be essential in an emergency situation.

Input :-
Employee ID of doctor.

Output :-

The method will display the contact number of doctor.

Parameters :-

Employee ID of doctor.

Exceptions :-

If the employee id isn't there in the database then an error message is displayed and end user is requested to enter the doctor's data before accessing doctor's contact no.

3.5.4.2.2 Methods : Display_specialization

Purpose :-

This method will display the specialization of doctor which may be essential to find the most suitable doctor in emergency situations.

Input :-

Employee ID of doctor.

Output :-

The method will display the specialization of the doctor.

Parameters :-

Employee ID of doctor.

Exceptions :-

If the employee id isn't there in the database then an error message is displayed and end user is requested to enter the doctor's data before accessing doctor's specialization.

3.5.5 Class : Administrator

3.5.5.1 Class Description

This class will help the user to give access rights to employees.

3.5.5.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
Integer	employeeId	private	NOT NULL	Contains the Employee Id of the administrator.
String	username	private	NULL	Contains the username of the administrator.
String	password	private	NULL	Contains the password of the administrator.

3.5.5.2.1 Methods : give_access_to_resources

Purpose :-

This method will allow administrator to give access to patient records, accounts information to doctors and receptionists depending upon their designations.

Input :-

Employee ID of doctor/Receptionist, designation.

Output :-

Success message is displayed in case access rights are given successfully.

Parameters :-

Employee ID of doctor/Receptionist, designation.

Exceptions :-

If the employee id isn't there in the database then an error message is displayed and end user is requested to enter the doctor's/receptionst's data before giving access to resources.

3.5.6 Class : Consultation Data

3.5.6.1 Class Description

This class will help the user to store the details of consultation data.

3.5.6.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
Date	Date of visit	private	NULL	Contains the date of visit of the patient.
String	Appointment time	private	NULL	Contains the appointment time of patient.
String	illness	private	NULL	Contains patient's illness.
String	symptoms	private	NULL	Contains patients' symptoms.
String	doctor	private	NULL	Contains name of doctor patient has consulted.
String	prescription	private	NULL	Contains patient's prescription.
String	test results	private	NULL	Contains patient's test results.
String	services	private	NULL	Contains services patient has rendered during his visit.

3.5.6.2.1 Methods : Display_last_visit

Purpose :-

This method will display the patient's last visit's information which will be helpful for the doctor in diagnosing a patient's illness and improvement over last visit .

Input :-

Patient ID.

Output :-

The method will display details of patient's last visit.

Parameters :-

Employee ID of doctor.

Exceptions :-

If the patient id isn't there in the database then an error message is displayed and end user is requested to enter the patient's data before accessing patient's last visit.

3.6 Module : Patient Dashboards

This functionality associates a dashboard to each patient which displays an overview of a patient's crucial clinical and personal information. to enable quick and efficient care. The

information to be displayed on the Patient Dashboard must include patient personal information (name, address, email id, phone number, blood group, medical history), diagnosis history, lab results, nutritional values, vitals, treatments, radiology documents.

The software must facilitate the user to create and configure multiple dashboards per patient, which in turn enables users to conveniently view information which can be filtered and organized based on their needs.

The dashboard must also portray graphical trends of numeric observations and a department specific view of patient clinical data.

3.6.1 Class : User

3.6.1.1 Class Description

This class represents the user of the system who can create, view and configure patient dashboards.

3.6.1.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	name	private	NULL	name of the user
String	employeeTYpe	private	NULL	type of employee : admin or front-desk staff
String	designation	private	NULL	designation of employee
String	employeeSSN	private	NULL	SSN of user
String	email	private	NULL	email id of user
String	phoneNumber	private	NULL	contact number of user

3.6.2 Class : Dashboard

3.6.2.1 Class Description

This class represents the component of the system which captures and maintains patient related information.

3.6.2.2 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	name	private	NULL	name of the patient
String	email	private	NULL	email id of user

String	phoneNumber	private	NULL	contact number of user
String	address	private	NULL	address of the patient
String	bloodgroup	private	NULL	blood group of patient
String	diagnosishistory	private	NULL	patient's medical history
String	medicalhistory	private	NULL	patient's diagnosis history
dictionary of key value pairs	Vitals	private	NULL	patients vitals entered as key value pairs when filled in relevant form during registration
dictionary of key value pairs	nutritionalvalues	private	NULL	patients nutritional indices entered as key value pairs when filled in relevant form during registration
blob	test result documents	private	NULL	patient's lab test results

3.6.3 Class : Receptionist

3.6.3.1 Class Description

It is a derived class from Person class. It is used to represent the Receptionist working for the hospital. It consists of all the attributes and methods of base class.

3.6.3.2 Method : create_dashboard

Purpose :-

This method is automatically invoked by the system when a new patient is registered.

Input :-

The input to the method includes all the registration details aforementioned and specified in the registration module..

Output :-

None, internally a unique patient dashboard would have been created

Parameters :-

The input to the method includes all the registration details aforementioned and specified in the registration module..

Exceptions :-

None

3.6.3.3 Method : view_dashboard

Purpose :-

It is used by the receptionist or doctor to view patient's dashboard.

Input :-

User should enter patient registration id corresponding to the required patient details

Output :-

The corresponding patient's dashboard is displayed

Parameters :-

Patient registration id

Exceptions :-

An invalid patient id would lead to an error message.

3.6.3.4 Method : configure_dashboard

Purpose :-

It is used by the receptionist or doctor to enter appointment/consultation data to the patient's dashboard.

Input :-

User should enter the patient registration id along with the parameters to be updated

Output :-

Success message in case the data gets committed into the database else retry message is displayed.

Parameters :-

All the required patient details to be added/updated along with the patient registration id

Exceptions :-

Retry message is displayed in case the patient data is not added in to the database.

4.0 Traceability Matrix

CRS Reference Section No. and Name	DESIGN / HLD Reference Section No. and Name	LLD Reference Section No. Name
Patient Registration - CRS Section 4.7-4.14	HLD Document Section - 3.7	3.2
In-Patient Management-CRS Section CRS Section 4.1-4.6	HLD Document Section - 3.2	3.1
Patient List - CRS Section 4.15	HLD Document Section - 3.4	3.3
Patient Dashboard - CRS Section 4.16	HLD Document Section - 3.5	3.6
Outpatient Management - CRS Section 4.17-4.22	HLD Document Section - 3.3	3.5
Patient Billing and Accounting - CRS Section 4.23-4.24	HLD Document Section - 3.6	3.4