

DENTAL MANAGEMENT

SYSTEM FOR DENTAL

CLINIC



EMILIA TITUS (214003051) (PROJECT MANAGER)

JASON OPERI UHONGORA (220124116)

GERALDO HELGARDO LIEBENBERG (220029326)

DAVID N. SAKARIA (201086263)

ERASTUS NAMAKASA (200934260)

Project Report**1. Title of Project**

Dental Management System for Dental Clinic

2. Objective of the Project

The Dental Clinic Management application allows for the access and consumption of information. Personal information, medical and dental histories, treatment plans, patient education, case reports, and other information are all included. This makes your dental visit more convenient, efficient, and precise, allowing for a paperless workplace. We are committed to communicating the proper method of control while providing insight into the many sorts of Dental Management systems. We understand that a Dental Management System isn't always a product to be sold; rather, it's a tool to be used to manage the internal operations of a dental facility by utilizing the appropriate gadget functions.

3. Project Category

RDBMS

4. Front End: HTML5, CSS, Bootstrap 5, FontAwesome 5, JavaScript, jQuery, J2EE

5. Back End: J2EE – 11, MariaDB 10, TomCat 10

6. Hardware Requirements

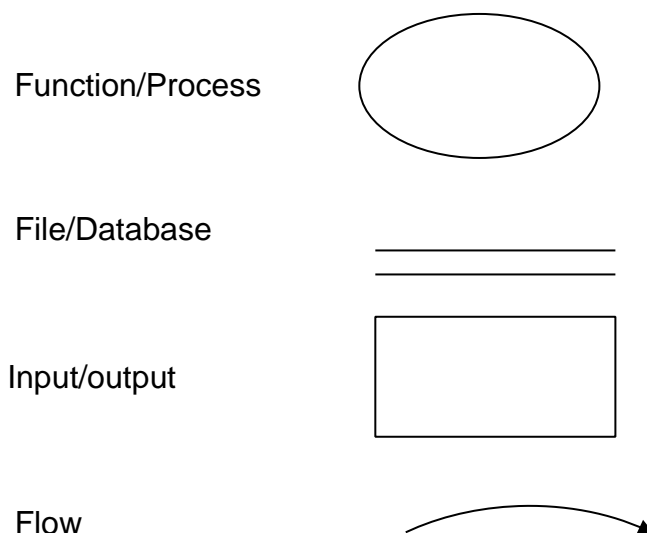
None – Cloud Hosted

Data - Flow Diagram

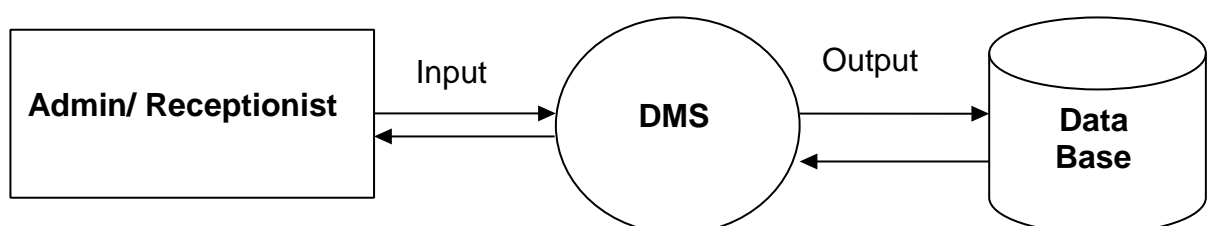
The Data Flow Diagram (DFD) is a graphical representation of information flowing through an Information System as it "goes with the flow." Data Processing can also be visualized

using a piece of information that goes with the drift diagram. It's not uncommon for a fashion designer to start with a conditions-stage DFD, which depicts the interaction between the device and the outside world. This DFD is then "exploded" to expose more elements of the gadget being simulated. A DFD is a device that allows you to follow the flow of data. Throughout the hassle analysis, data and drift graphs are commonly employed. A DFD denotes the movement of data through the device's various versions or ways. Dataflow diagrams can be utilized to show the stop person how the information they enter influences the device's overall shape, from order to dispatch to restocking. The proper sign-in information is kept and updated in a database by the appropriate authorities.

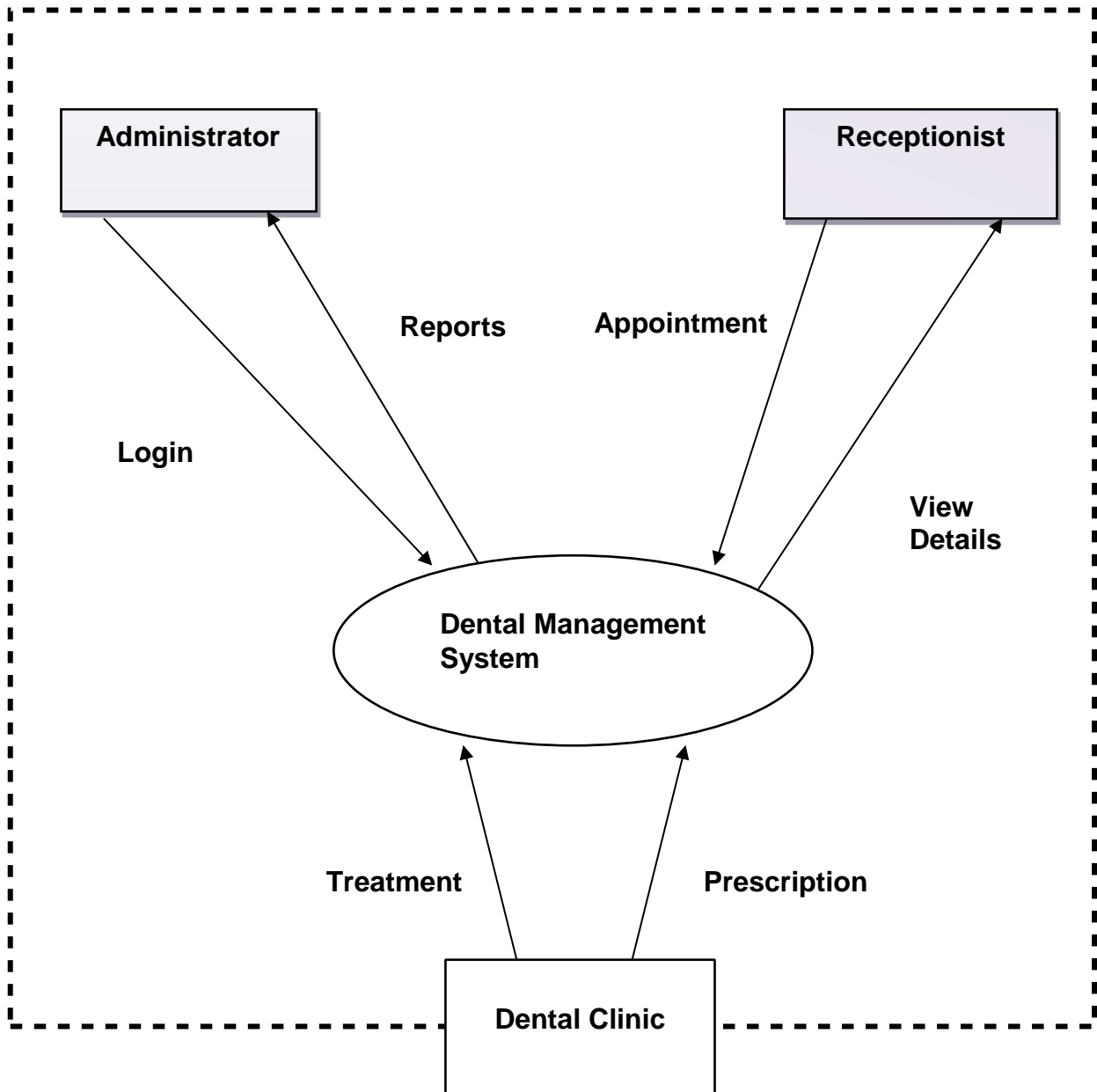
Data Flow Diagram Notation



Level 0



Level 1



MODULE DESCRIPTION

The System has different pages for various activities.

Patient Personal Details

This page is where you may get information on the patient. The options provided here are used to capture all the patient's information.

Appointment

This page is used to keep track of Daily Patient Appointments for a certain period of time in the consultation process.

Treatment

This page is used to store treatment details for each patient, along with dates, tooth numbers, and more.

Prescription

This page deals with activities related to keeping a patient's prescription, such as dosage and days.

Case Reports

The purpose of this module is to display the complete patient's medical history, prescription, and data for treatment.

Admin

Includes master table data permissions and some special permissions.

ER - Diagram

The ER Diagram is a specialized graphic that depicts the interrelationships between database elements. ER diagrams frequently utilize symbols to represent three distinct types of data. Entities are usually represented by boxes. Diamonds are frequently used to represent relationships, while ovals are used to represent characteristics. In software program engineering, an Entity Relationship Model (ERM) is a summary and conceptual explanation of statistics. Entity Relationship Modelling (ERM) is a relational schema database modelling method that is used to provide a top-down conceptual schema or semantic statistics version of a system, usually a relational database.

Entity

An entity is something about which you want to keep track of data with. This is the foundation for storing information about business processes. An entity is a type of object that is defined in an information system that stores data. Within a firm, there are various entities.

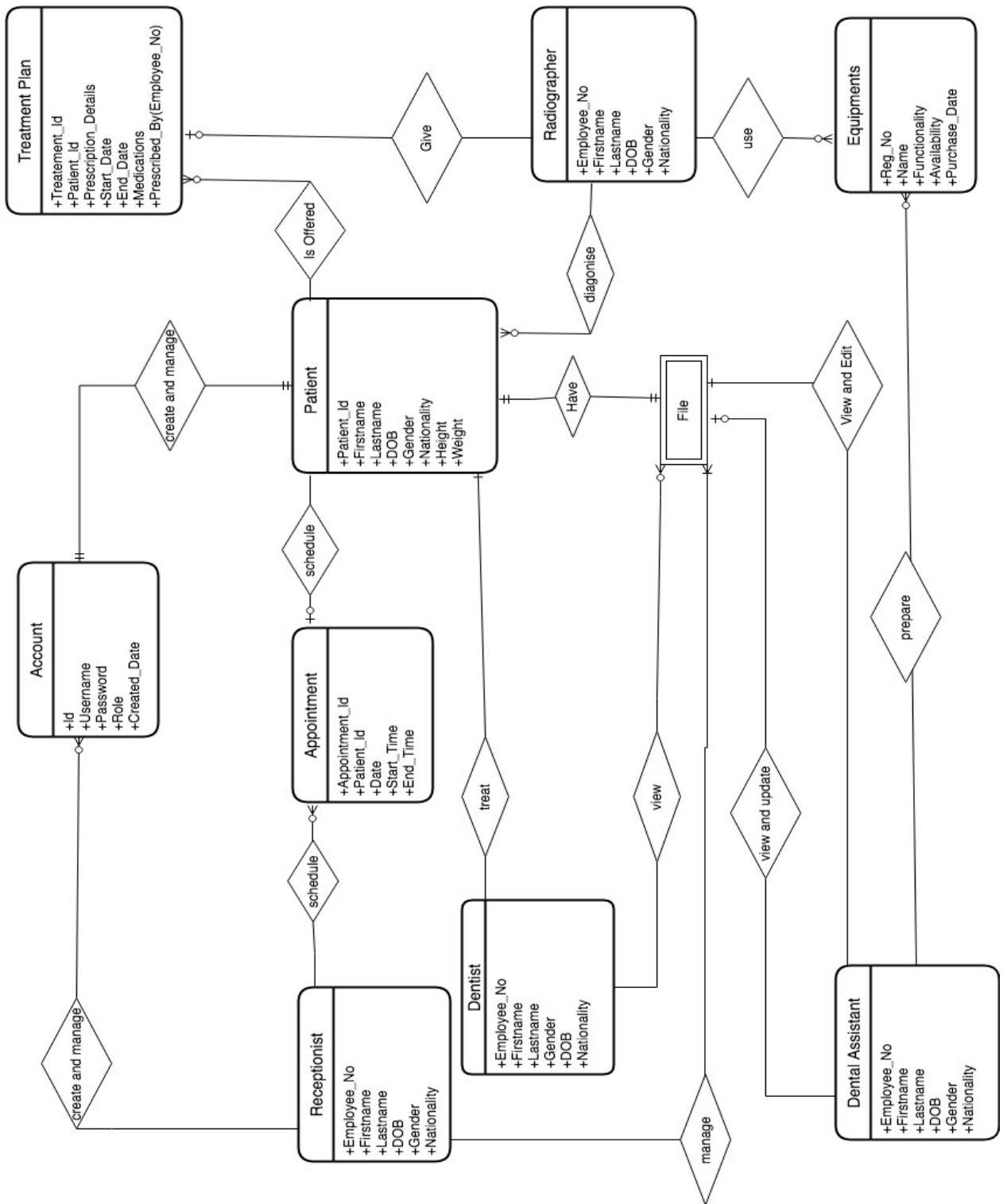
Relationships

Relationships are named collections or connections between entities, or they are used to connect two or more entities with a few commonplace qualities or significant interactions among objects.

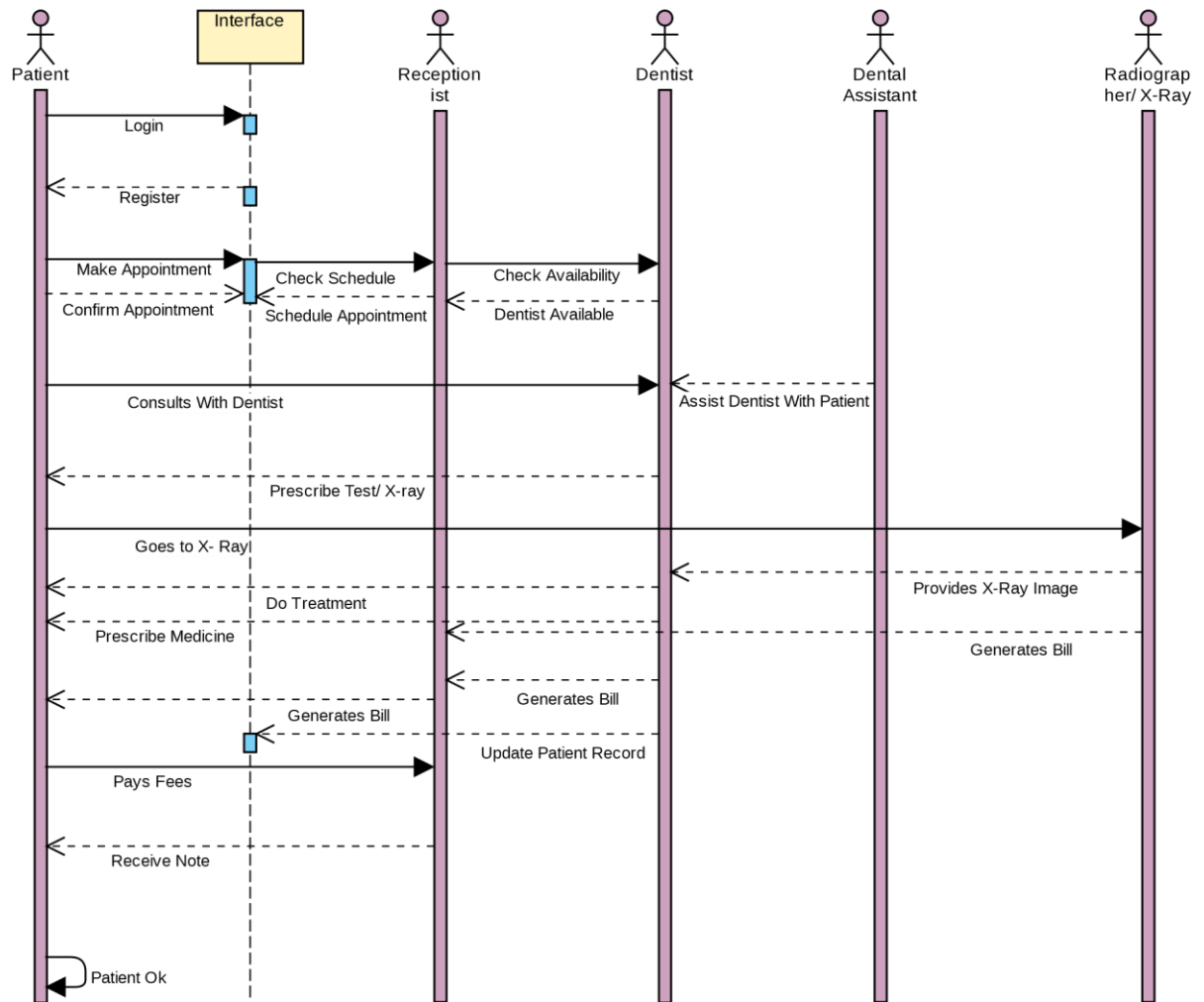
Attributes

Entity properties, entity relationships, and descriptors are all examples of attributes. Attributes are the fundamental characteristics of an entity.

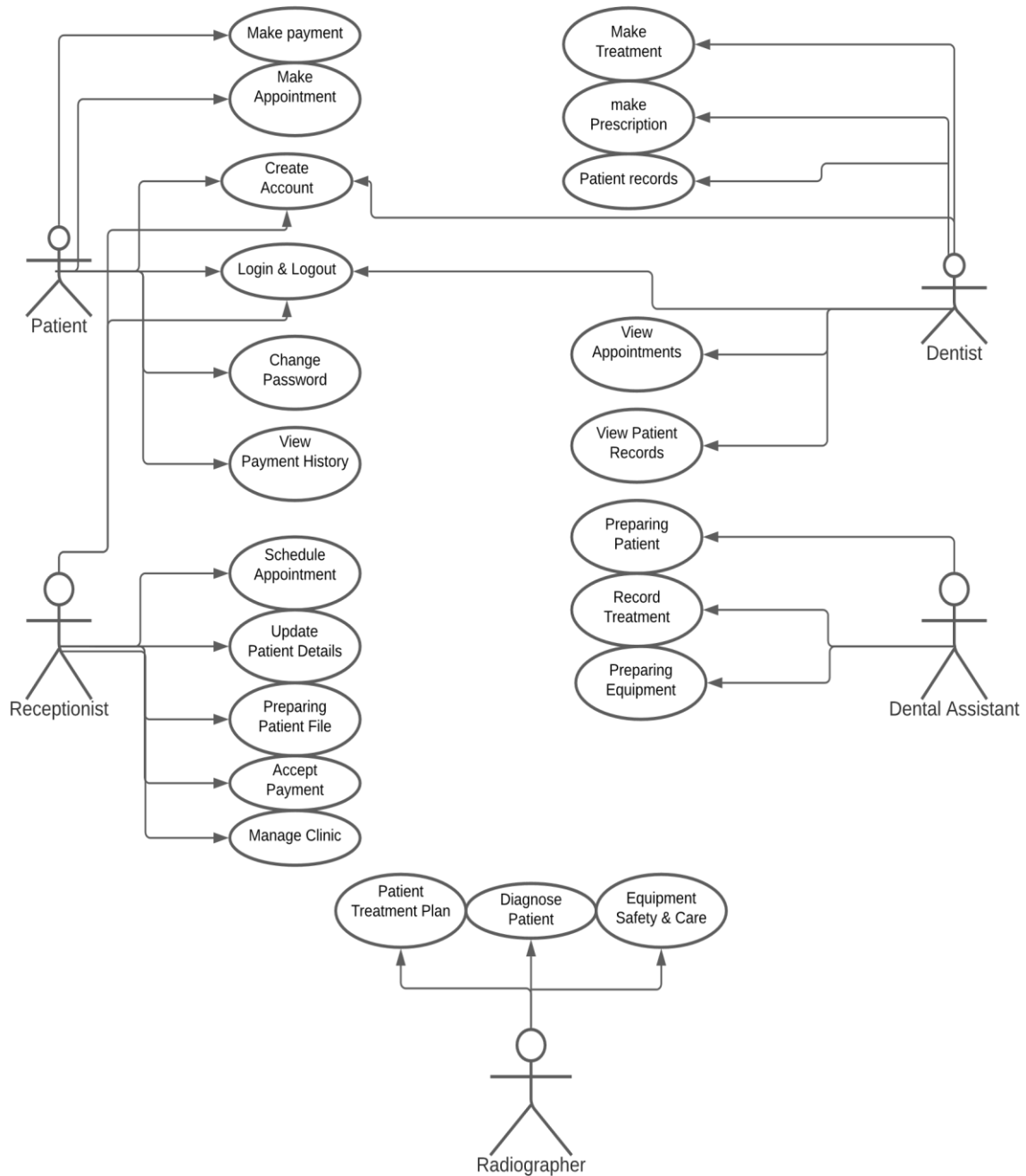
ER Diagram of DMSC



Sequence Diagram



Use Case Diagram



Testing

INTRODUCTION

The process of executing a program to detect bugs is known as testing. By identifying system design deviations and mistakes, testing increases system integrity. The goal of the

test is to pinpoint locations where mistakes are most likely to occur. This aids in the avoidance of system errors. Testing adds value to the invention by ensuring that it meets the needs of the user.

Testing is mostly used to detect regions that are prone to system flaws and faults. The test should be comprehensive and well-organized. A system that has been partially tested is just as harmful as one that has not been tested at all. Untested systems are also more costly. The last and most crucial stage is execution. Testing entails putting a system built with various forms of data to the test. When a method is tested, an error is discovered and the method's correctness is recorded.

GOALS OF TESTING

The goal of testing are:

- Testing is the procedure of running a program to find bugs.
- Successful test cases are test cases that reveal bugs that have not yet been discovered.

System testing is a component of the implementation process that ensures the system is working correctly and effectively in accordance with user requirements before it goes live. As previously said, testing is crucial to the success of your system. System testing logically believes that if all pieces of the system are accurate, the aim will be achieved successfully. Before the system is ready for user testing, it goes through a series of tests.

TESTING METHODS

The execution phase is when the system is put to the test. This is to guarantee that your system is fully functional and capable before going live. Testing is critical to the system's success. Online reaction, capacity, strain, recovery, safety, and usability testing are all performed on applicant systems. To prepare the chosen system for user acceptance testing, a series of tests are done.

The Testing Steps are

Unit Testing

The focus of unit tests is on the smallest unit of software program design. This is referred to as page testing. Individual pages are put to the test. The test is carried out at the programming stage. This phase validates that each page is functioning properly in relation to the module's expected output.

Integration Testing

The interface has the potential to lose data. One module can have a negative impact on another. You won't be able to link in the main function in the desired way if you combine the sub-functions. While building a software structure, integration testing is a systematic way to perform tests to uncover problems in an interface. The purpose is to create a program form and get a unit tested module. All of the pages are mixed together and examined as a whole.

Validation

The program is fully packed, which is a highlight of integration testing. The final set of software tests begins with validation tests, which are used to find and resolve interface issues. Validation tests can be described in a variety of ways, but the most straightforward definition is that validation is successful if the program performs as expected by the client. There are three possible outcomes after the validation test is completed.

- The overall performance attributes, or characteristic traits, meet the requirements and are accepted.
- A deviation from specification is exposed and a deficiency lists is created.
- Proposed system under attention has been examined with the aid of using the use of validation take a look at and located to be operating acceptable.

Output Testing

Following the validation test, the suggested system output test is performed.

This is because the system will be ineffective unless it generates the appropriate output in a specific configuration. I discovered that the output format displayed on the screen is correct. When creating the system, the format was created to meet the needs of the user. Also available in physical copy. The output is customized to the user's specific requirements. As a result, the output test failed to fix the system.

User Acceptance Testing

The user's acceptance of the system is critical to the system's success. By maintaining in touch with future system users during development and making changes as needed, the system under consideration tests user approval.

Executed in accordance to this

- Input Screen Design
- Output Screen Design
- Online communication to guide user
- Format of statements and alternative outputs

Security mechanisms

Authentication is built into the system, and those who do not have it will be denied access. As a result, only authorized users have access to the software. The system will be vulnerable to outsiders if lawful users reveal their credentials.

7. Project Scope:

The Application database can be hosted online with a purpose to control all departments at once.

CONCLUSION

The project titled “**Dental Management System**” is developed using HTML5, CSS, Bootstrap 5, FontAwesome 5, JavaScript, jQuery, J2EE as front end and J2EE – 11, MariaDB 10, GlassFish Ver. 6.1.0 in back-end to computerize the process of inventory and sales system.

Many features are already integrated into this project. The main beneficiary is the "**Dental Management System**", which spends a lot of time collecting patient data. In addition, additional attributes can be recognized and integrated in the near future.