

System &Design

**Online Doctor Appointment System**

**Submitted To**

**Dr. Ebrahim Mansoor**

**Ghazala Shafi**

**GitHub Repository Link**

[**https://github.com/rshammad/S.E-PROJECT**](https://github.com/rshammad/S.E-PROJECT)

**Group Members Name:**

***1. HAMMAD AZAM RAEES (13542)*** [***Raeeshammad68@gmail.com***](mailto:Raeeshammad68@gmail.com)

***2. S.M YAWAR ABBAS (13651)*** [***Abbasaskari90@gmail.com***](mailto:Abbasaskari90@gmail.com)

***3. HADEER UR REHMAN (12413)*** [***Hadeerehman@gmail.com***](mailto:Hadeerehman@gmail.com)

***4. DANISH AJAZ AHMED (13309)*** [***Danishajaz34@gmail.com***](mailto:Danishajaz34@gmail.com)

***5. AGHA AHMED SHAYAN (13855)*** [***Aghashayan50@gmail.com***](mailto:Aghashayan50@gmail.com)

**Project Introduction:**

## The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor’s appointment online. This is a web based application that overcomes the issue of managing and booking appointments according to user’s choice or demands.

## The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time.

**PROJECT OBJECTIVE**

Aim of this project is that to enhance the relationship between a doctor and patient and this project will help patients to book appointment of doctor and also help the user to send their report and view their medical progress. The system allows doctor to manage their booking slots online. Patients are allowed to book empty slots online. This system manages the appointment schedule. The system manages the appointment data for multiple doctors of various dates. Each time a user visits the portal doctor his/her medical entry is stored in the database by admin. Next time when the user login. The history is shown to the patient and patient search for doctor which doctor is online get start communication and if he confirms the request for the appointment initiated a mail is send by system for the scheduled date of doctor and patient data and send the day on which day is schedule

**Project Plan & System Design:**

This subsection contains the requirements for the. Online doctor appointment system. These requirements are organized by the features discussed in the vision document. Features from vision documents are then refined into use case diagrams and to sequence diagram to best capture the functional requirements of the system. All these functional requirements can be traced using tractability matrix

* **FR01**: The user should be able to register and manage his appointments online at any time.
* **FR02**: Database has to store all the information efficiently without any information loss.
* **FR03:** The user shall be able to search for the doctors by specialty, name, hospital and by area or city.
* **FR04**: The user can change his profile info at any time
* **FR05**: Doctors can manage all appointments made with him on his account
* **FR06**. The system shall notify the user about any conflict in the current configuration
* **FR07**. The system shall maintain users email information as a required part of users profile.
* **FR08**. The system shall send a confirmation to the user through email.
* **FR09**.The system shall display the online appointment that are eligible to change.
* **FR10**. The system shall display the reviews and ratings of each doctor, when it is selected.
* **FR11**.The system shall enable the user to enter their reviews and ratings.

**Non-functional requirement**

**• Portability requirements:**

o PR01: A website has to be compatible with different popular web browsers

(Google Chrome, Mozilla Firefox, Opera, Safari and Internet Explorer 8+) • Reliability requirements:

o RR01: The probability of failure less than 0.01%

o RR02: Uptime of at least 99%

o RR03: Less than 30 minutes needed to recover from system failure.

**PROJECT PHASES**

Since every software project is different, there is no set list of deliverables that every project must provide. Part of your task is to decide what is needed for this specific project. Typical deliverables include working code, documentation, training materials, etc.

Following assignments must be completed and submitted on the provided deadlines during the course. These are group projects, but you will be given individual grades for parts of some assignments.

|  |  |  |  |
| --- | --- | --- | --- |
| **Assignment** | **Deliverables** | **Duration** | **Deadlines** |
| 1 | Project Proposal (Report) | 1 week |  |
| 2 | Feasibility study and Project Plan (Report) | 2 weeks |  |
| 3 | Requirements Specification Document (Presentation & Documentation) | 1 week |  |
| 4 | Design Document (Presentation & Documentation)  • User Interface Design  • System & Program Design | 2 weeks |  |
| 5 | Unit test scripts, test summaries (Report) | 1 weeks |  |
| 6 | Final delivery (Presentation & Documentation) | 1 weeks |  |

# PROCESS TO BE FOLLOWED

For this project, the team has decided to follow Water fall approach that involves beginning with a user requirement and adding functionality until all of the client’s requirements are met.

**3.0 Methodology**

* Waterfall Model



**Justification of Methodology**

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* The project is short.
* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model. each phase has specific deliverables and a review process.
* Phases are processed and completed one at a time.
* Easy to arrange tasks.

**Description of Methodology**

The sequential phases in Waterfall model are:

* **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
* **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
* **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
* **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
* **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

**OUTLINE PLAN**

**I. Milestone (Requirement Analysis)**.

An initial design of the requirements analysis should be done as Milestone 1. This should come after a formal discussion with the Client.

**II. Milestone 2 (Requirements Analysis (final)).**

The final draft of the requirements analysis should be done for Milestone 2. In addition, a presentation will be prepared as a part of this milestone.

**III. Milestone 3– Software Architecture and Design.**

An initial draft of the software architecture and design should be done as Milestone 3. A meeting with the Client should follow Milestone 3 to get feedback on the design of the system.

**IV. Milestone 4 – Software Architecture and Design (final).**

A final draft of the software architecture and design document should be done for Milestone 4. A presentation should be prepared for the Client.

**V. Milestone 5 (Database)**

The database is an essential part of the system, as it is the center of all information. All following system components depend on this deliverable. A database schema needs to be fixed for Milestone 5 to provide a basis for the other components to be based on.

**VI. Milestone 7 (Map and Menu).**

The map and the menu are the front-end graphical web interface that the public user sees and interacts with. Milestone 7 is to reach feature-completion on the requirements.

**VII. Milestone 8 (Testing, Debugging, and Integration).**

The system needs to be well-tested, debugged at this milestone. Once the system has passed the acceptance test, it needs to be integrated into the actual production system for this milestone.

**IX. Milestone 9 (Project Deadline)**

## The project source code should be handed over to the Client for the final milestone. A presentation is presented to the Client

**Functional and Non-Functional Requirements**

The following is the desired functionality of the new system.

Accept of submissions in form of raw patients; perform analysis of financial to authenticate the users of the system. And non-functional requirement includes the following

The system must verify the validate all user input ant user must be notified in case of errors detected in the database, the system should allow room for expansion.

**Entity Relationship (E-R) Diagram**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. An entity relationship diagram is a means of visualizing how the information a system produces is related.

**Entity**

Which are represented by rectangle. An entity is an object or concept that has its existence in the real world. It includes all those things about which data is collected. A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

**Attributes**

Which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

**An Entity Set**

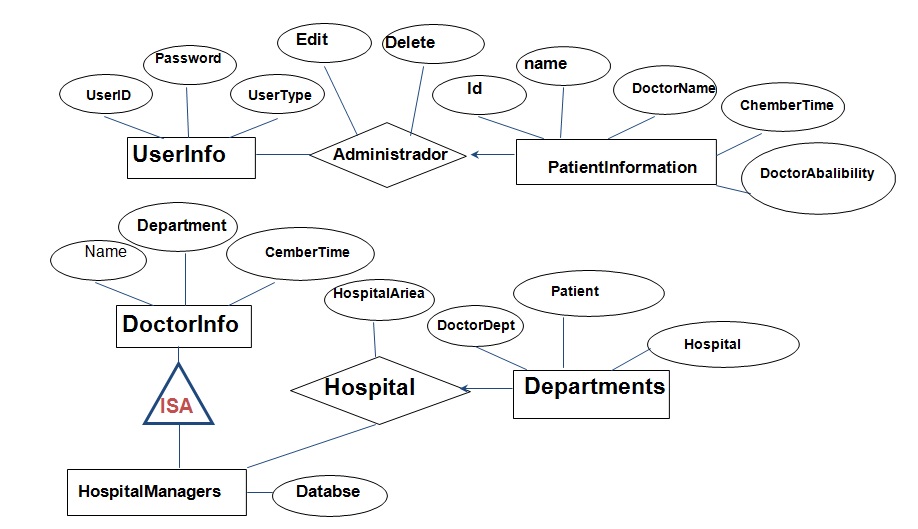
It is a set of entities of the same type that share the same properties, or attributes.

**Process**

A process shows a transformation or manipulation of data flows within the system.

**Actions**

Which are represented by diamond shapes, show how two entities share information in the database.



**Use case diagram**

