

Online Health Information System

BD Doctor

A Thesis

Submitted to the Department of Computer Science and Engineering

of

BRAC University

by

Md. Jahangir Hossain (ID # 02201143)

A.T.M.Mahmudul Hasan (ID # 02201090)

In Partial Fulfillment of the

Requirements for the Degree

of

Bachelor of Science & Engineering

June 2006

ACKNOWLEDGMENTS

Special thanks to our supervisor Matin Saad Abdullah who was always available for us to give directions. Mr. Abdullah was so helpful to us that it would be really impossible to publish this thesis paper without his enormous help. During our thesis we had to search several native and international websites for collecting information. We specially give thanks to www.doctorsofbangladesh.com, <http://neuro-www2.mgh.harvard.edu/hospitalwebusa.html>, for giving us so much information. Here we also need to pay gratitude to Naila miss to help us whenever we need any kind of information and help. She was really helpful to us all the times. We also give thanks to some hospital, especially their websites. Because we have got several ideas such as interface and content and resources from their websites. Finally we pay our tribute to almighty Allah for everything we could come up with.

Table of Contents.

1. Abstract	IV
2. Introduction	V
3. Project description	V
4. Project Implementation	VI
4.1. Requirement	VI
1.1.1. Methodology	VI
Use case	VI
Scope and goals of a use case	VIII
Degree of detail	IX
Use case templates.....	X
Use case name	X
Iteration	X
Summary	XI
Preconditions	XI
Triggers	XI
Basic course of events.....	XI
Alternative paths.....	XI
Postconditions	XII
Business rules	XII
Notes	XII
Author and date	XII
Use cases and the development process	XII
Benefits of Use Cases	XIII
Limitations of Use Cases	XIV
1.1.2. Implementation	XIV
5. Design	XVII
5.1. Methodology	XVII
Relational database.....	XVII
Relational model	XVIII
Mapping the relational model to a relational database	XVIII
Database normalization	XIX
Relational algebra	XIX
Schematic example of a relational database	XX
6. Conclusion	XXII
7. Appendix.....	XXIII

1. Abstract

First world countries are adopting IT in their daily life extremely. Especially medical services are extremely qualified. They can search in the net about their health related problems; Even latest technologies, services and medicine. On the net they can even search for their best affordable doctors and hospitals. They can get admitted into it, they can make appointment over net. After all it can be said that most of the countries in the world are using IT for the medical sector as a powerful tools of giving services to the people so that they get the best service. Unfortunately, in Bangladesh we don't have enough health related services. What's more! What we have is not known to the general people. General people don't know we have world class medical care for brain tumor and that like diseases. The main goal of BD Doctor is to make people know that we have the good services in medical sector. BD Doctor will make people aware by giving information of health.

To design and develop such a thesis paper we had to search the current health system and we also had to search for information into the web so that we can propose the international qualities in the service of health in Bangladesh.

2. Introduction

It is impossible to lead a peaceful life without a sound health. However, to have a sound health we need a well-organized health care system. In our Bangladesh, we have very good facilities in the medical sector. But, due to less advertisement or information most of the people are not aware of it. Moreover, to find the exact doctor is not too easy. Sometimes we may find a doctor in a far place but we don't know there is a doctor just next to our door. Therefore, if it could be possible that anyone can search for the best doctor or health facilities in any given quarry then it would be beneficiary. Another important thing is the cost. If it is possible to search for the optimal point of cost and services among the service providers (hospital or clinic) then it will be a plus point for the general people. Sometimes medical specialists come from foreign country, special discount given to a particular health service, special discount given to health care devices but this kind of health related advertisement not collected together so that people get the benefit of it. If it could be the case that some media or website combine all the things together for the better service of health it would be excellent. We are proposing such a web based health care system where people will get services of all these ideas and many more.

3. Project description

For giving the description of the project initially we need to say why we have come up with BD Doctor Idea? It is really a matter of sorrow that doctor people communication is not too easy still now. And there is not a well trusted friendly relation in between them. Still now if we have any serious health problem at the first thinking we plan to go to India, Singapore, and Thailand. However we don't even know that at a cheap cost in Bangladesh we have that solution, not only the solution but also the standard quality solution. Here is another problem in this sector. Consider the scenario, a person come from a rural area to the town for taking appointment. Unfortunately it is very tough to get the service in the very first day. That particular person needs to wait for several hours to get a serial of another day. So, he gets back to village and come back that day to get the service. What we see is lake of communication system in between doctor and patient.

BD Doctor gives information and services in almost all the ways of medical sector. The main goal of BD Doctor is to spread out the information and services of medical sector even to the root level. List of the services given by BD Doctor is described bellow:

- I. **Doctor Information:** To know the health sector we need to know the key conductor, i.e. Doctor of this area. BD Doctor provides a huge

amount of doctor's information with the necessary related features of a doctor. That is why finding the exact doctor that we need is extremely easy! Just put your key features like specialty, region, fees of a doctor and BD Doctor Database will show you a list of doctors as you have given the criterion. Based on this information people can select the best doctor at the optimal point of quality, fees, service and distance.

- II. Hospital Information:** In the extreme case, we need to go to the hospital. This is a rare case that people visit hospital for the services and facilities when they are not ill. Usually when people become ill, they find a hospital to get admitted as early as possible. However, in a hurry you might not get into the exact hospital you are looking for. However, BD Doctor gives the largest number of hospital and clinic's information. From this information, it will be very easy to find out the desired hospital. One can find out the desired hospital or clinic by giving several criterions.
- III. Make Appointment:** BD Doctor provides the facility to give an appointment to a particular hospital, clinic or chamber in a particular date and time. No matter where you are, just come into BD Doctor site and place your appointment. However, BD Doctor System will verify this appointment system so that the possibilities of keeping the appointment for the patient become high. In our manual current system, it is very tough to have an appointment to a particular doctor. Physically we need to go there and have to get a serial number for a particular day. However, in BD Doctor this process will be extremely easy. Just browse the doctor's profile, hospital and clinic and depending on your needs place an online appointment by simply some mouse click!
- IV. Access Calendar:** BD Doctor provides a well-featured calendar for both the doctor and the patient. A doctor can see how many appointments he has in a particular day. Moreover, this calendar is also capable of storing personal schedules. A patient can see his appointment to a particular doctor and his personal schedules. In the calendar there is short notes given. If appointment and schedule is clicked then details of appointment and schedule will be displayed.

4. Project Implementation

4.1. Requirement

1.1.1. Methodology

Use case



BD Doctor

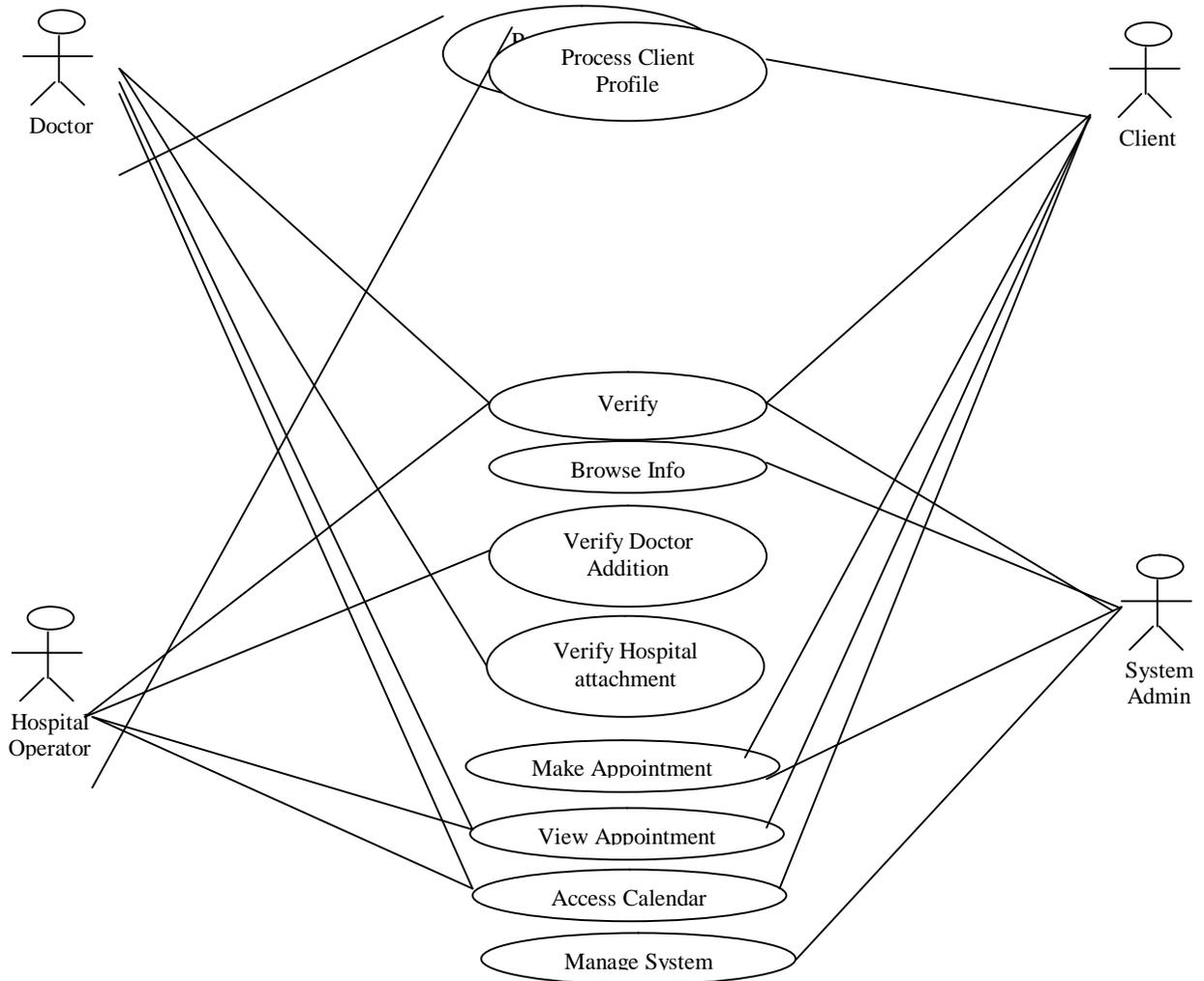


Fig: Use Cases of BD Doctor

In software engineering, a **use case** is a technique for capturing the potential requirements of a new system or software change. Each use case provides one or more *scenarios* that convey how the system should interact with the end user or another system to achieve a specific business goal. Use cases typically avoid technical jargon, preferring instead the

language of the end user or *domain expert*. Use cases are often co-authored by Business Analysts and end users.

In 1986, Ivar Jacobson, an important contributor to the Unified Modeling Language (UML) and Rational Unified Process, captured the concept of the use case. Originally he had written *usage scenarios* but then came up with the Swedish word *användarfall* (on-ven-dor-fahl) which roughly means "situation of usage" or "usage case" but usage case did not sound good in English so instead he wrote Use Case. Jacobson's codification of the idea was influential and seminal. Numerous contributions have been made to the subject since then, but the most significant, influential and comprehensive, in terms of defining what use cases are and how to write them effectively, was provided by Alistair Cockburn, in his 2000 book *Writing Effective Use Cases*.

During the 1990s use cases became one of the most common practices for capturing functional requirements. This is especially the case within the object-oriented community where they originated, but their applicability is not restricted to object-oriented systems, because use cases are not object oriented in nature.

Scope and goals of a use case

Each use case focuses on describing how to achieve a single business goal or task. From a traditional software engineering perspective a use case describes just one feature of the system. For most software projects this means that multiple, perhaps dozens, of use cases are needed to embrace the scope of the new system. The degree of formality of a particular software project and the stage of the project will influence the level of detail required in each use case.

A use case defines the interactions between external actors and the system under consideration to accomplish a business goal. Actors are parties outside the system that interact with the system; an actor can be a class of users, roles users can play, or other systems.

Use cases treat the system as a *black box*, and the interactions with the system, including system responses, are as perceived from outside the system. This is a deliberate policy, because it simplifies the description of requirements, and avoids the trap of making assumptions about how this functionality will be accomplished.

A use case should:

- describe a business task to serve a business goal
- have no implementation-specific language
- be at the appropriate level of detail
- be short enough to implement by one software developer in a single *release*.

Degree of detail

Alistair Cockburn in *Writing Effective Use Cases* identifies three degrees of detail in writing use cases...

Brief

A brief use case consists of a few sentences summarizing the use case. It is highly suitable to use a spreadsheet for planning software development. A brief use case can be easily inserted in a spreadsheet cell, and allows the other columns in the spreadsheet to record business priority, technical complexity, release number, etc

Casual

A casual use case consists of a few paragraphs of text, covering the items mentioned above, elaborating the use case in the form of a summary or story.

Detailed

A detailed or complex use case is a formal document based on a long template with fields for various sections; and it is the most common understanding of the meaning of a use case. Detailed use cases are discussed in detail in the next section on use case templates.

Appropriate detail

Some software development processes do not require anything more than a simple use case to define requirements. However, some other development processes require detailed use cases to define requirements. The larger and more complex the project, the more likely that it will be necessary to use detailed use cases.

The level of detail in a use case often differs according to the progress of the project. The initial use cases may be brief, but as the development process unfolds the use cases become ever more detailed. This reflects the different requirements of the use case. Initially they need only be brief, because they are used to summarise the business requirement from the point of view of users. However, later in the process, software developers need far more specific and detailed guidance.

Rational Unified Process invites developers to write a brief use case description in the diagram, with a casual description as comments and a detailed description of the flow of events in a textual analysis. All those can usually be input into the UML software, or can be written separately in a text editor.

Use case templates

There is no standard template for documenting detailed use cases. There are a number of competing schemes, and individuals are encouraged to use templates that work for them or the project they are on. Standardization within each project is more important than the detail of a specific template. There is, however, considerable agreement about the core sections; beneath differing terminologies and orderings there is an underlying similarity between most use cases.

Typical sections include:

- Use Case Name
- Iteration
- Summary
- Preconditions
- Triggers
- Basic course of events
- Alternative paths
- Postconditions
- Business rules
- Notes
- Author and date

Different templates often have additional sections, e.g. assumptions, exceptions, recommendations, technical requirements. There may also be industry specific sections.

Use case name

The use case name provides a unique identifier for the use case. It should be written in verb-noun format (e.g., *Borrow Books*, *Withdraw Cash*), should describe a completable goal (e.g., *Register User* is better than *Registering User*) and should be sufficient for the end user to understand what the use case is about. Goal-driven use case analysis will name the use cases according to the Actor's goals thus ensuring use cases are strongly user centric. Two to three words is the optimum length. If four or more words are proposed in the name there is usually a shorter better name that can be used.

Iteration

Often an iteration section is needed to inform the reader the stage a use case has reached. The initial use case developed for business analysis and scoping may well be very different from the evolved version of that use case when the software is being developed. Older versions of the use case may still be current documents, because they may be valuable to different user groups.

Summary

The *summary* section is used to capture the essence of the use case before the main body is complete. It provides a quick overview which is intended to save the reader from having to read the full contents of a use case to understand what the use case is about. Ideally a summary is just a few sentences or a paragraph in length and includes the goal and principal actor.

Preconditions

A *preconditions* section is used to convey any conditions that must be true when a user initiates a use case. They are not however the triggers that initiate a use case. Where one or more preconditions are not met, the behaviour of the use case should be considered indeterminate.

Triggers

The *Triggers* section describes the starting condition(s) which cause a use case to be initiated. Can be external, temporal or internal.

Basic course of events

At a minimum, each use case should convey a *primary scenario*, or the typical course of events. The main basic course of events is often conveyed as a set of usually numbered steps, for example:

1. The system prompts the user to logon.
2. The user enters his name and password.
3. The system verifies the login information.
4. The system logs user on to system.

...and so on.

Alternative paths

Use cases may contain secondary paths, or alternative scenarios which are variations on the main theme. Exceptions, or what happens when things go wrong, may also be described, either within the alternative paths section or in a section on their own. The alternative paths make use of the numbering of the basic course of events to show at which point they differ from the basic scenario, and if appropriate where they rejoin. The intention is to avoid repeating information unnecessarily.

An example of an alternative path would be:

1. The system recognizes cookie on users machine.
2. Go to step 4 (Main path)

An example of an exception path would be:

3. The system does not recognize users logon information
4. Go to step 1 (Main path)

According to Anthony J H Simons and Ian Graham, alternate paths were not originally part of use cases. Instead, each use case represented a single user's interaction with the system. Or in other words, each use case represented one possible path through the system. Multiple use cases would be needed before designs based on them could be made. In this sense, use cases are for exploration, not documentation.

Postconditions

The *post-conditions* section summarizes the state of affairs after the scenario is complete.

Business rules

Business rules are written or unwritten rules that determine how an organization conducts its business with regard to a use case. Business rules are a special kind of assumption. Business rules may be specific to a use case or apply across all the use cases, and indeed all the business.

Notes

Experience has shown that whatever template is used, analysts discover there is always important information that doesn't fit the structure of the template. Therefore each template usually includes a section for such seemingly inevitable information.

Author and date

This section should list when this version of the use case was created and who documented it. It should also list and date any versions of the use case from an earlier stage in the development which are still current documents. The author is traditionally listed at the bottom, because it is not considered to be essential information; use cases are intended to be collaborative endeavors and they should be jointly owned.

Use cases and the development process

The specific way use cases are used within the development process will depend on which development methodology is being used. In certain development methodologies, a brief use case survey is all that is required. In other development methodologies, the use cases evolve in complexity and change in character as the development process proceeds. In some methodologies, they may begin as brief business use cases, evolve into more

detailed system use cases, and then eventually develop into highly detailed and exhaustive test cases.

Benefits of Use Cases

Use cases are a newer, more agile format for capturing software requirements. They are often contrasted to large, monolithic documents that attempt but fail to completely convey all possible requirements before construction of a new system begins.

Use cases have a number of advantages:

- Use case modeling (including the writing of use case specifications) is generally regarded as an excellent technique for capturing the functional requirements of a system.
- Use cases discourage premature design.
- Use cases are traceable.
- Use cases can serve as the basis for the estimating, scheduling, and validating effort.
- Use cases are reusable within a project. The use case can evolve at each iteration from a method of capturing requirements, to development guidelines to programmers, to a test case and finally into user documentation.
- Use case alternative paths capture additional behaviour that can improve system robustness.
- Use cases are useful for scoping. Use cases make it easy to take a staged delivery approach to projects; they can be relatively easily added and removed from a software project as priorities change.
- Use cases have proved to be easily understandable by business users, and so have proven an excellent bridge between software developers and end users.
- Use case specifications don't use a special language. They can be written in a variety of styles to suit the particular needs of the project.
- Use cases allow us to tell stories. It is very easy to describe a use case in a concrete way by turning it into a story or scenario.
- Use cases are concerned with the interactions between the user and the system. They make it possible for user interface designers to become involved in the development process either before or in parallel with software developers.
- Use cases put requirements in context; they are clearly described in relationship to business tasks.
- Use case diagrams help stakeholders to understand the nature and scope of the business area or the system under development.
- Use case diagrams can be recorded using the UML and maintained within widely available CASE tools
- Use cases and use case diagrams can be fully integrated with other analysis and design deliverables created using a CASE tool to produce a complete requirements, design and implementation repository
- Test Cases(System, User Acceptance and Functional) can be directly derived from the use cases

Limitations of Use Cases

Use cases are not without their limitations:

- Use cases are not well suited to easily capturing non-functional requirements of a system.
- Use cases templates do not automatically ensure clarity. Clarity depends on the skill of the writer(s).
- Use cases are not well suited to safety critical and real time systems where a greater degree of precision is required.
- Use cases have a learning curve involved in interpreting them correctly, for both end users and programmers.
- Proponents of Extreme Programming often consider use cases too needlessly document-centric, preferring to use the simpler approach of a User story.
- Use case developers often find it difficult to determine the level of user interface (UI) dependency to incorporate in the use case. While use case theory suggests that UI not be reflected in use cases, many find it awkward to abstract out this aspect of design as it makes the use cases difficult to visualize.

1.1.2. Implementation

Process Doctor Profile:

Primary Actor:

Doctor

Stakeholder and interest:

Doctor:

Using the doctors ranking system s/he can see the condition of his/her reputation. Client can be increased based on the reputation. Doctors can maintain his daily schedule by using the site calendar. Doctors can give his/her service easily and quickly to the client. Reduce advertising cost. Sending broker to collect client is not required.

Hospital:

Reputation will be increased. Hospital information and doctors information belongs to the hospital will easily be reached to the client by the site. Hospital advertisement takes place by the site.

Client:

Client can easily get appointment by using this site. Client can easily search for any specialized. Hospital/chamber information can easily be known to the client. Client can take proper decision to get better service by observing doctors ranking, hospital/chamber information. No afraid to fall on brokers bad experience.

Precondition:

Doctor should be verified, Database should be updated.

Post condition:

If successful, then new doctor information will be added in the database system and doctor can change his profile by securely login.

Main Success scenario:

1. Doctor successfully enters into the web site.
2. Registration form comes.
3. Doctor gives his name.
4. Doctor gives his registration number which is given by the Bangladesh medical council.
5. Doctor gives the user ID.
6. Doctor gives the password.
7. Doctor confirms the given password.
8. Doctor gives his picture.
9. Doctor selects sex.
10. Doctor writes down degree.(repeat)
11. Doctor writes the name of the institution from where s/he got the degree.(repeat)
12. Doctor notes down regarding his degree.(repeat)
13. Doctor writes his specialization.(repeat)
14. Doctor writes the name from where s/he got the specialization.(repeat)
15. Doctor notes down regarding his specialization.(repeat)
16. Doctor gives the Experience.
17. Doctor gives the chamber name.(repeat)
18. Doctor gives maximum number of patient he would visit per day (repeat).
19. Doctor gives the chamber address (repeat).
20. Doctor gives the chamber phone number (repeat).
21. Doctor gives the chamber mobile number (repeat)
22. Doctor gives the URL of the chamber (repeat).
23. Doctor gives the visiting hour (repeat)
24. Doctor gives first visiting fee (repeat).
25. Doctor gives second visiting fee (repeat).
26. Doctor gives the hospital name.(repeat)
27. Doctor gives the hospital address (repeat).
28. Doctor gives the designation. (repeat).
29. Doctor gives the room number (repeat).
30. Doctor gives the hospital phone number (repeat).
31. Doctor gives the hospital mobile number (repeat)
32. Doctor gives the URL of the hospital (repeat).
33. Doctor gives the visiting hour (repeat)
34. Doctor gives first visiting fee (repeat).
35. Doctor gives second visiting fee (repeat).
36. Doctor gives personal mobile number.

37. Doctor gives personal phone number.
38. Doctor gives personal Fax number.
39. Doctor gives personal Email.
40. Doctor gives personal URL.
41. Doctor selects the secret question.
42. Doctor writes the secret answer.
43. After inserting necessary information doctor will go to the next step by clicking next button for further processing.
44. Hospital verification process held here by matching the information given by both doctor and hospital operator.
45. System will send the information to the particular hospital to be verified if any request wouldn't come from any hospital operator.
46. After all the steps the information will be stored in the database.

Extension:

- *a At any time system fails.
1. Doctor restarts the system.
 - 2a. If the registration form not loads successfully press F5 or Ctrl+F5.
 - 3a. If doctor does not fill the name field then he/she will prompt to give the name again.
 - 4a. If doctor does not fill the registration number field then he/she will prompt to give the registration number again.
 - 5a. If doctor does not fill the USER ID field then he/she will prompt to give the USER ID again.
 - 5b. If the user ID is not unique then system will prompt the user for giving unique user ID or System can propose unique ID.
 - 6a. If doctor does not fill the password field then he/she will prompt to give the password again.
 - 7a. If doctor does not fill the confirm password field then he/she will prompt to fill the field again.
 - 10a. If the degree field is not filled then he/she will prompt to fill the field again.
 - 11a. If the institution name field is not filled then he/she will prompt to fill the field again.
 - 13a. If the doctor doesn't fill the specialization field then he/she will prompt to fill the field again.
 - 14a. If the doctor doesn't fill the specialization place field then he/she will prompt to fill the field again.
 - 16a. If the doctor doesn't fill the experience field then he/she will prompt to fill the field again.
 - 17a. If chamber name field is not filled then he/she will prompt to fill the field again.
 - 18a. If the doctor doesn't give the maximum number patient he visits per day then he/she will prompt to fill the field again.

- 19a. If chamber address field is not filled then he/she will prompt to fill the field again.
- 20a. If chamber phone number field is not filled then he/she will prompt to fill the field again.
- 21a. If chamber mobile number field is not filled then he/she will prompt to fill the field again.
- 23a. If the doctor doesn't give the visiting hour then s/he will prompt to give the visiting hour again.
- 24a. If the doctor doesn't give first visiting fee then s/he will prompt to give the first visiting fee again.
- 25a. If the doctor doesn't give second visiting fee then s/he will prompt to give the second visiting fee again.
- 26a. If hospital name field is not filled then s/he will prompt to fill the filled again.
- 27a. If hospital address field is not filled then s/he will prompt to fill the filled again.
- 28a. If designation field is not filled then s/he will prompt to fill the filled again.
- 29a. If room number field is not filled then s/he will prompt to fill the filled again.
- 30a. If phone number field is not filled then s/he will prompt to fill the filled again.
- 31a. If mobile number field is not filled then s/he will prompt to fill the filled again.
- 33a. If visiting hour field is not filled then s/he will prompt to fill the filled again.
- 34a. If first visiting fee field is not filled then s/he will prompt to fill the filled again.
- 35a. If second visiting fee field is not filled then s/he will prompt to fill the filled again.
- 41a. If secret question field is not filled then s/he will prompt to fill the filled again.
- 42a. If secret answer field is not filled then s/he will prompt to fill the filled again.
- 43a. If next page is not shown for further processing then s/he will start from step-1 again.
- 44a. If the verification process is not complete then again start from step-1.
- 45a. If the information sending process is not complete then again start from step-1.
- 46a. If storing process is not complete then again start from step-1.

5. Design

5.1. Methodology

Relational Database

Relational database

A **relational database** is a database structured in accordance with the relational model. Strictly speaking the term refers to a specific collection of data but it is invariably employed together with the software used to manage that collection of data. That software is more correctly called a relational database management system (RDBMS). Relational database management systems incorporate many features from the relational

model, but commercial RDBMSs also tend to diverge from the relational model in significant ways.

Edgar Codd, who proposed the relational model in a groundbreaking 1970 paper, introduced a set of rules and, in 1985, a scoring system to determine whether or not a particular database implementation was in fact "relational." By this standard most commercial database servers fail the test, although it's worth noting that there have been few implementations of "truly" relational database servers and these have not had anywhere near the same commercial success. This article will explain what a database which conforms to the relational model is and is not. See relational database management system for an examination of how real-world database server software functions.

Relational model

The **relational model** for database management is a data model based on predicate logic and set theory. Relational databases have features which correspond with features of the relational model. In order to make the software work well in the real world, relational database management systems add additional features such as indices which have no place in the relational model.

Mapping the relational model to a relational database

The core of the relational model is that all data can be organized into relations. Relations are a way of organizing the data which permit flexible and powerful operations on the stored values. A relation has two parts, the **heading**, which is called the metadata in a relational database, and the **body**, which is the data itself. Relational databases generally implement relations in the form of tables and views.

A relation's heading is a set of **attributes**, and an attribute consists of a name and a domain. An attribute in relational theory is usually implemented as a column in a relational database. The domain is the set of possible values which can be stored. For example, a relation could have an attribute called "Percentage" with a domain of the integers between 0 and 100, inclusive. This would most commonly be implemented in a relational database as a column with a name of "Percentage" and a domain that has the data type of integer and a constraint disallowing values outside of that range. Note, however, that the data type and the constraint are really just two different ways of limiting the possible values which can be stored; from a relational theory point of view they're doing essentially the same thing, which is describing the possible attribute values.

One area where relational database management systems tend to diverge from relational theory is that RDBMSs usually give their columns an order. For example, they have a "first column," which might be the primary key, and a "second column," which might be a name, and so on. This is required by the SQL standard but is proscribed by relational theory. Most real-world attributes don't have orders, either. If you were to describe a person you wouldn't say that their eye color comes "before" their height and "after" their name, but this is essentially what the SQL standard requires.

The body of a relation is a set of unordered tuples. A tuple is a collection of values, organized by attribute name rather than by a defined order. For example, we might say that the attribute value of the "Percentage" attribute for a particular tuple is 95. The tuples are unordered in two different senses: The attribute values they contain are identified by name rather than by a "position" within the tuple, and the tuple itself has no single position relative to other tuples. Tuples are generally implemented in relational databases as rows. As noted earlier, most RDBMSs diverge from the relational model in that they are required by SQL to have an order of columns, but it's reasonably common for a database server to not support a defined order for their rows, and this seeming limitation turns out to have a number of advantages both in theory and practice.

The relational model (and Codd's rule number four) requires that the metadata also be accessible in the same manner as the rest of the data, and nearly all RDBMSs support this rule by describing the metadata in "system tables." Hence, the heading and the body are distinct within a single relation, but are essentially the same thing in general.

Database normalization

Main article: Database normalization

Database normalization is the process of structuring real-world data in such a way that it meets the standards for a relational database.

Relational algebra

Main article: Relational algebra

The relational algebra is a set of operations that manipulate relations as they are defined in the relational model and as such describes part of the data manipulation aspect of this data model. Because of their algebraic properties these operations are often used in database query optimization as an intermediate representation of a query to which certain rewrite rules can be applied to obtain a more efficient version of the query.

The exact set of operations may differ per definition and also depends on whether the unlabeled relational model (that uses mathematical relations) or the labeled relational model (that uses the labeled specialization of mathematical relations) is used. We will assume the labeled case here as this is the most common way the relational model is defined. That means that we assume that tuples are partial functions from attribute names to values. The value of the tuple t on the attribute a is denoted in this article as $t(a)$.

A k -adic relation is a set of k -tuples that constitutes the extension of a k -adic predicate. However, these tuples differ from the more abstract tuples of mathematics by having more concrete qualities associated with the places of the relation. In this setting, the components of the tuples, called attribute values or feature values, are identified by means of attribute names or feature names. Queries and integrity constraints are expressed declaratively, without the use of iterative loops or pointers, using operators based on the relational algebra and relation comparisons. The relational algebra is complete with respect to first-order predicate calculus except that restrictions are imposed

on the use of the logical operations of negation and disjunction in order to guarantee that database computations will be feasible in practice.

Schematic example of a relational database

A concept of relation that suffices to begin can be set out as follows.

- Defined in extension, a k -adic relation L is a set of k -tuples.
- A k -tuple \mathbf{X} is a sequence of k elements, x_1, \dots, x_k , called the components of the k -tuple. The components of the k -tuple \mathbf{X} are commonly indicated by writing either one of the following two forms:

$$\mathbf{X} = (x_1, \dots, x_k)$$

$$\mathbf{X} = x_1 : \dots : x_k$$

It is critically important to understand that a relation, considered, as one says, in extension, is a set, in other words, an aggregate entity or a collection of things. Thus a k -tuple is not a relation, it is only an element of a relation, what is naturally enough called an elementary relation.

Table 1 shows how the k -tuples of a k -adic relation might be conceived in tabular form, with the k -tuple $\mathbf{X}_i = (x_{ij})_{j=1}^k = (x_{i1}, \dots, x_{ik}) = x_{i1} : \dots : x_{ik}$ supplying the entries for the i^{th} row of the Table.

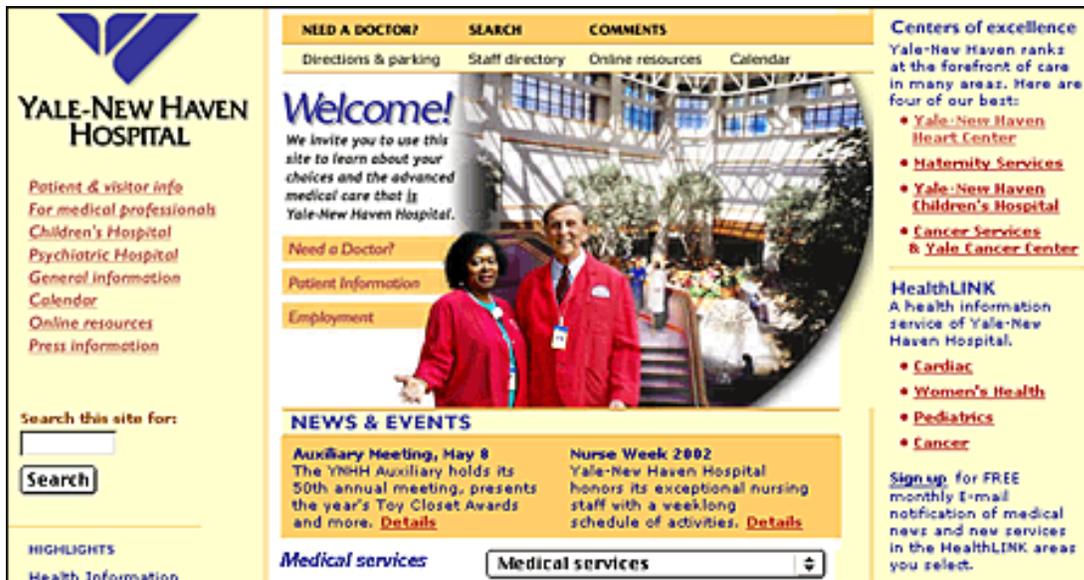
Domain 1	Domain 2	...	Domain j	...	Domain k
X_{11}	X_{12}	...	X_{1j}	...	X_{1k}
X_{21}	X_{22}	...	X_{2j}	...	X_{2k}
...
X_{i1}	X_{i2}	...	X_{ij}	...	X_{ik}
...
X_{m1}	X_{m2}	...	X_{mj}	...	X_{mk}

Web design

Web design is a crucial part of a website. The design of the site involves following things.

- ❖ The design of the site will determine its organizational framework
- ❖ What your audience wants from you
- ❖ What you wish to say and...
- ❖ How to arrange the content to best meet your audience's needs

❖ Ask yourself what your audience wants, and center your site design on their needs. To show the contents, home page should contain link of the all major things of the site. For example following site can be considered.



For organizing information following things should be considered. There are five basic steps in organizing your information:

- 1) Divide your content into logical units
- 2) Establish a hierarchy of importance among the units
- 3) Use the hierarchy to structure relations among units
- 4) Build a site that closely follows your information structure
- 5) Analyze the functional and aesthetic success of your system

Information in a web page should be chunked in meaningful topics. And need to consider following things.

- Few Web users spend time reading long passages
- Discrete chunks of information lend themselves to Web links
- Chunking can help organize and present information in a uniform format
- Concise chunks of information are better suited to the computer screen

A basic information structure of a website is given bellow

- The simplest way to organize information is to place it in a sequence
- Sequential ordering may be
 - Chronological
 - A logical series of topics progressing from the general to the specific
 - Alphabetical, as in indexes, encyclopedias, and glossaries

Adding what's new page is a good idea to catch up viewer's attention.

- Many Web sites need to be updated frequently so that the information they present doesn't become stale.
- Place a "new" graphic next to each updated item.
- However, your site is complex, with many levels of information spread over dozens (or hundreds) of pages, you might consider making a "What's New" page

To communicate with viewer's you need to have such a system so that people can contact with you and can give you their opinions. To do so you can consider the following thing.

- The Web is a bidirectional medium — people expect to be able to send you comments, questions, and suggestions.
- Always provide at least one link to an email address in a prominent location in your site.
- You can request user information and feedback using Web page forms and then use a database to store and analyze their input.

In a website viewers may have several question related with you site. You can sort out the general type of question and can have a frequently asked question's answer page named FAQ. To design a FAQ page consider following things.

- FAQ (frequently asked questions) page, is the most commonly asked questions from users are listed along with answers
- Most questions from new users have been asked and answered many times before
- A well-designed FAQ page can improve users' understanding of the information and services offered and reduce demands on your support staff.

Implementation
examples
Construction
Screenshots
Testing
Methodology
Testcases
Implementation
examples

6. Conclusion

Bangladesh is still so much backward in medical side. When internationally medical services are getting more standard day by day, we are not concern about it. No such actions are taken to give better service to this sector. We need to be more conscious about it. There are million of people who are searching for better health treatment but not

getting it. What's more unfortunately they are being missed treated some times. We have shown several ways of improving the service in medical area. Much more research is needed in this field. We hope the service will be international standard day by day.

7. Appendix

Use cases

Process Hospital Info:

Primary Actor:

Hospital Operator

Stakeholder and interest:

Doctor:

Using the doctors ranking system s/he can see the condition of his/her reputation. Client can be increased based on the reputation. Doctors can see the hospital facilities at any time and can suggest client for proper diagnosis and facilities in the hospital. Doctors can give his/her service easily and quickly to the client. Reduce advertising cost. Sending broker to collect client is not required.

Hospital:

Reputation will be increased. Hospital information and doctors information belongs to the hospital will easily be reached to the client by the site. Hospital advertisement takes place by the site.

Client:

Client can easily get appointment by using this site. Client can easily search for any specialized and hospital facilities. Hospital information can easily be reached to the client harassment. Client can take proper decision to get better service by observing different hospital information and facilities. No afraid of fall on brokers bad experience.

Precondition:

Hospital should be verified, Database should be updated.

Post condition:

If successful, then new hospital information will be added in the database system and hospital operator can change the profile by securely login.

Main Success scenario:

1. Hospital operator successfully enters into the web site.
2. Registration form comes.
3. Hospital operator gives hospital name.

4. Hospital operator gives address.
5. Hospital operator gives phone number.
6. Hospital operator gives mobile number.
7. Hospital operator gives Fax number.
8. Hospital operator gives Email.
9. Hospital operator gives URL.
10. Hospital operator gives out door fee.
11. Hospital operator gives out door time.
12. Hospital operator gives number of cabin.
13. Hospital operator gives number of word.
14. Hospital operator gives cabin charge per day.
15. Hospital operator gives bed charge per day.
16. Hospital operator gives service name. (repeat)
17. Hospital operator gives fee of the specified service. (repeat)
18. Hospital operator writes down notes. (repeat)
19. Hospital clicks on the button to add doctor's profile.
20. Hospital operator will be given a form to add doctor's profile. (repeat)
21. Hospital operator gives doctor name. (repeat)
22. Hospital operator gives doctor registration number given by Bangladesh Medical Council. (repeat)
23. Hospital operator gives designation of the doctor. (repeat)
24. Hospital operator gives designation of the doctor. (repeat)
25. Hospital operator gives room number of the doctor. (repeat)
26. Hospital operator gives phone of the doctor. (repeat)
27. Hospital operator gives first visiting fee of the doctor. (repeat)
28. Hospital operator gives second visiting fee of the doctor. (repeat)
29. Hospital operator gives starting service date of the doctor. (repeat)
30. After entering all information of the doctor click on the button OK to send the information to the doctor to be verified.
31. System will send the information to be verified.
32. After verification all necessary information will be stored in the database.

Extension:

- *a At any time system fails.
1. Hospital Operator restarts the system.
 - 2a. If the registration form doesn't not load successfully press F5 or Ctrl+F5 to reload.
 - 3a. If Hospital Operator does not fill the hospital name field then he/she will prompt to give the name again.
 - 4a. If hospital operator doesn't give hospital address then s/he will prompt to fill the filled again.
 - 5a. If hospital operator doesn't give hospital phone number then s/he will prompt to fill the filled again.
 - 6a. If hospital operator doesn't give hospital mobile number then s/he will prompt to fill the filled again.

- 10a. If hospital operator doesn't give hospital out door fee then s/he will prompt to fill the filled again.
- 11a. If hospital operator doesn't give hospital out door time then s/he will prompt to fill the filled again.
- 12a. If hospital operator doesn't give number of cabin then s/he will prompt to fill the filled again.
- 13a. If hospital operator doesn't give number of word then s/he will prompt to fill the filled again.
- 14a. If hospital operator doesn't give cabin charge per day then s/he will prompt to fill the filled again.
- 15a. If hospital operator doesn't give bed charge per day then s/he will prompt to fill the filled again.
- 16a. If hospital operator doesn't give service name then s/he will prompt to fill the filled again.
- 17a. If hospital operator doesn't give fee of the specified service then s/he will prompt to fill the filled again.
- 19a. If hospital operator doesn't click on the button to add doctors profile then s/he will be notified to click on the button.
- 20a. If the form doesn't load then again start from step-19.
- 21a. If hospital operator doesn't give doctor name then s/he will prompt to fill the filled again.
- 22a. If hospital operator doesn't give doctor registration number given by Bangladesh Medical Council then s/he will prompt to fill the filled again.
- 23a. If hospital operator doesn't give doctors designation then s/he will prompt to fill the filled again.
- 24a. If hospital operator doesn't give doctors room number then s/he will prompt to fill the filled again.

- 25a. If hospital operator doesn't give doctors phone number then s/he will prompt to fill the filled again.
- 26a. If hospital operator doesn't give doctors first visiting fee then s/he will prompt to fill the filled again.
- 27a. If hospital operator doesn't give doctors second visiting fee then s/he will prompt to fill the filled again.
- 28a. If hospital operator doesn't give starting service date of the doctor then s/he will prompt to fill the filled again.
- 30a. If the system is not able not send the information to be verified then again start from step-29.

Process Client Info:

Primary Actor:

Client

Stakeholder and interest:

Doctor:

As the doctors ranking is evaluated by the client hence using the doctors ranking system doctor can see the condition of his/her reputation. Client can be increased based on the reputation. Doctors can give his/her service easily and quickly to the client.

Hospital:

Reputation will be increased. Hospital information and doctors information belongs to the hospital will easily be reached to the client by the site. Hospital advertisement takes place by the site.

Client:

Client can easily get appointment by using this site. Client can easily search for any specialized. Hospital/chamber information can easily be known to the client. As a result reduce harassment of finding specialized. Client can take proper decision to get better service by observing doctors ranking, hospital/chamber information. No afraid to fall on brokers bad experience.

Precondition:

Client should be verified, Database should be updated.

Post condition:

If successful, then new Client information will be added in the database system and client can change the profile by securely login.

Main Success scenario:

1. Client successfully enters into the web site.
2. Registration form comes.
3. Client gives his name.
4. Client selects sex.
5. Client gives the user ID.
6. Client gives the password.
7. Client confirms the given password.
8. Client gives phone number.
9. Client gives mobile number.
10. Client gives Email.
11. Client gives address.
12. System will send the information to the system admin to be verified.
13. After verification all necessary information will be stored in the database.

Extension:

- *a At any time system fails.
 - 1. Client restarts the system.
 - 2a. If the registration form doesn't not load successfully press F5 or Ctrl+F5.
 - 3a. If client does not fill the name field then he/she will prompt to give the name again.
 - 5a. If client does not fill the USER ID field then he/she will prompt to give the USER ID again.
 - 5b. If the user ID is not unique then system will prompt the user for giving unique user ID or System can propose unique ID.
 - 6a. If client does not fill the password field then he/she will prompt to give the password again.
 - 7a. If doctor does not fill the confirm password field then he/she will prompt to fill the field again.
 - 7b. If password and confirm password doesn't match then s/he will prompt to fill password and confirm password field again.
 - 8a. If client doesn't give phone number then s/he will prompt to fill the filled again.
 - 9a. If client doesn't give mobile number then s/he will prompt to fill the filled again.
 - 11a. If client doesn't give address then s/he will prompt to fill the filled again.
 - 12a. If system cant send the information to be verified then again start from step-1.
 - 13a. If system cant store the information in the database then again start from step-1.

Process Make Appointment

Primary Actor:

Client

Stakeholder and interest:

Doctor:

Doctor's available visiting information and advertisement takes place here. Doctor can give his/her service easily and quickly to the client. Doctor can easily see his booking schedule and information by using the site.

Hospital:

Hospitals all kinds of booking visiting information and advertisement take place here. As a result hospital can reach those kinds of information to the clients easily. Reputation will be increased. Hospital information and doctors information belongs to the hospital will easily be reached to the client by the site. Hospital advertisement takes place by the site.

Client:

Client can easily get appointment by using this site. Client can easily give booking for any medical service. Hospital/chamber information can easily be known to the client. As a result reduce harassment of booking. Client can take proper decision to get better service by observing doctors hospital/chamber booking schedule. No afraid of fall on brokers bad experience.

Precondition:

Client should be verified and securely login into the system, Database should be updated.

Post condition:

If successful, then booking information will be added and stored in the database system.

Main Success scenario:

1. Client successfully enters into the web site to login.
2. Client writes the user ID in the user ID field.
3. Client gives the password in the password field.
4. Client press OK button to login.
5. System match user ID given by the client with user ID stored in the database.
6. System match user password given by the client with user password stored in the database.
7. If user ID and password match with the user ID and password stored stored in the database then client will be logged in securely.
8. Client selects Hospital Booking or Chamber Booking.
9. If Client selects Hospital Booking all necessary hospital information will be shown to the client.
10. If Client selects chamber booking all necessary chamber information will be shown to the client.
11. Client selects date.
12. System checks that the selected date is full of booking or not.
13. Client selects visiting Hour.
14. Client selects OK button to confirm visit.
15. After confirmation, booking information will be sent to the specified chamber or hospital and stored in the database.
16. Client is given a visiting number.

Extension:

*a At any time system fails.

1. Client restarts the system.
- 2a. If Client doesn't write user ID in the user ID field then s/he will prompt to fill the filled again.
- 3a. If Client doesn't write user password in the user password field then s/he will prompt to fill the filled again.
- 7a. If user ID and password doesn't match with the user ID and password stored in the database then client will not be logged in securely and prompt to fill the filled again.
- 9a. If Client doesn't select Hospital Booking then all necessary hospital information will not be shown to the client and prompt to select the filled again.
- 10a. If Client doesn't select chamber Booking then all necessary chamber information will not be shown to the client and prompt to select the filled again.
- 11a. If Client doesn't select date then s/he will prompt to fill the filled again.

12a. If selected date is full of booking then show another date. If not then go to next step.

13a. If Client doesn't select visiting Hour then will prompt to select visiting hour.

Process Verify Doctor Addition

Primary Actor:

Hospital Operator

Stakeholder and interest:

Doctor:

After verification real doctors are added into the site. As a result doctors can be benefited by using the site in different dimension.

Hospital:

After verification real doctors belonging to the hospital are added into the site. As a result the hospital is prevented from the use unauthorized persons.

Client:

Client can get the services of real hospitals and doctors.

Precondition:

Doctor should add profile into the site and system must send the information given by doctor to the Hospital Operator.

Post condition:

If successful, then doctor's profile will be added and stored in the database.

Main Success scenario:

1. Hospital operator successfully enters into the web site to login.
2. Hospital operator writes the user ID in the user ID field.
3. Hospital operator gives the password in the password field.
4. Hospital operator press OK button to login.
5. System match user ID given by the Hospital operator with user ID stored in the database.
6. System match user password given by the Hospital operator with user password stored in the database.

7. If user ID and password match with the user ID and password stored in the database then Hospital operator will be logged in securely.
8. Hospital operator will see the requests came from different doctors.
9. If the doctors are belonging to the hospital then Hospital operator verified the doctor.
10. Doctor's profile will be added and stored in the database.

Extension:

*a At any time system fails.

1. Hospital operator restarts the system.
- 2a. If Hospital operator doesn't write user ID in the user ID field then s/he will prompt to fill the filled again.
- 3a. If Hospital operator doesn't write user password in the user password field then s/he will prompt to fill the filled again.
- 7a. If user ID and password doesn't match with the user ID and password stored in the database then Hospital operator will not be logged in securely and prompt to fill the filled again.
- 9a. If the doctors are not belonging to the hospital then the hospital operator wouldn't verify the doctors request.

Process Verify Hospital Attachment

Primary Actor:

Doctor

Stakeholder and interest:

Doctor:

After verification real doctors are added into the site. As a result doctors can be benefited by using the site in different dimension.

Hospital:

After verification real doctors belonging to the hospital are added into the site. As a result the doctors are prevented from the use unauthorized persons.

Client:

Client can get the services of real hospitals and doctors.

Precondition:

Hospital operator should add the doctor's profile into the site and system must send the profile given by the hospital operator to the doctor.

Post condition:

If successful, then Hospital attachment will be added and stored in the database.

Main Success scenario:

1. Doctor successfully enters into the web site to login.
2. Doctor writes the user ID in the user ID field.
3. Doctor gives the password in the password field.
4. Doctor press OK button to login.
5. System match user ID given by the Doctor with user ID stored in the database.
6. System match user password given by the Doctor with user password stored in the database.
7. If user ID and password match with the user ID and password stored in the database then Doctor will be logged in securely.
8. Doctor will see the requests came from different Hospital operator.
9. If the hospital attachment came from Different hospital operator are valid then Doctor will verify the Hospital attachment.
10. Hospital attachment will be added and stored into the database.

Extension:

*a At any time system fails.

1. Doctor restarts the system.
- 2a. If Doctor doesn't write user ID in the user ID field then s/he will prompt to fill the filled again.
- 3a. If Doctor doesn't write user password in the user password field then s/he will prompt to fill the filled again.
- 7a. If user ID and password doesn't match with the user ID and password stored in the database then Doctor will not be logged in securely and prompt to fill the filled again.
- 9a. If Hospital attachments are not valid then the hospital operator wouldn't verify the hospital attachments.

Process View Appointment

Primary Actor:

Client, Doctor, Hospital operator.

Stakeholder and interest:

Doctor:

Doctor's can see the appointment by browsing this site and s/he can easily observe the statistics of the patients come to take his service per day.

Hospital operator:

Hospital operator can see the appointment by browsing this site and s/he can easily observe the statistics of the patients come to take the service of a particular doctor. By watching the appointment (which is updated every time responding the clients request)

operator can easily give information about appointment to the clients those who are willing to take appointment by phone or directly going to the chamber.

Client:

Client can easily see appointment by using this site. Client can easily take appointment for a particular doctor by viewing the appointment information. After making an appointment client can easily view the appointment given by him to make sure that appointment has been made successfully.

Precondition:

Client should securely login into the system. Doctor should securely login into the system. Hospital operator should securely login into the system. Appointment Database should be updated.

Post condition:

If successful, then appointment schedule will be available to the clients.

Main Success scenario:

1. Doctor/client/hospital operator successfully enters into the web site to login.
2. Doctor/client/hospital operator writes the user ID in the user ID field.
3. Doctor/client/hospital operator gives the password in the password field.
4. Doctor/client/hospital operator press OK button to login.
5. System match user ID given by Doctor/client/hospital operator with user ID stored in the database.
6. System match user password given by Doctor/client/hospital operator with user password stored in the database.
7. If user ID and password match with the user ID and password stored in the database then Doctor/client/hospital operator will be logged in securely.
8. Doctor/client/hospital operator clicks on the appointment field to see appointment schedule.
9. Appointment schedule page successfully load.
10. Doctor/client/hospital operator sees the appointment.

Extension:

*a At any time system fails.

1. Doctor/client/hospital operator restarts the system.
- 2a. If Doctor/client/hospital operator doesn't write user ID in the user ID field then s/he will prompt to fill the filled again.
- 3a. If Doctor/client/hospital operator doesn't write user password in the user password field then s/he will prompt to fill the filled again.
- 7a. If user ID and password doesn't match with the user ID and password stored in the database then Doctor/client/hospital operator will not be logged in securely

and prompt to fill the fields again.
9a. If Appointment schedule page doesn't load successfully then starts from step-8.

Process Access calendar

Primary Actor:

Client, Doctor, Hospital Operator, System admin.

Stakeholder and interest:

Doctor:

Doctor can make his daily routine by accessing the calendar. Doctor can see appointment schedule and personal notes on a particular day and can set the remainder for a particular event on a particular day at a specific time. The reminder sends a message to his/her mobile number on that day at a specific time.

Hospital operator:

Hospital operator can make his daily routine by accessing the calendar. Hospital operator can see his daily routine on a particular day and can set the remainder for a particular event on a particular day at a specific time. The reminder sends a message to his/her mobile number on that day at a specific time.

Client:

Client can make his daily routine by accessing the calendar. Client can see his daily routine on a particular day and can set the remainder for a particular event on a particular day at a specific time. The reminder sends a message to his/her mobile number on that day at a specific time.

System Admin:

System Admin can make his daily routine by accessing the calendar. System Admin can see his daily routine on a particular day and can set the remainder for a particular event on a particular day at a specific time. The reminder sends a message to his/her mobile number on that day at a specific time.

Precondition:

Client should securely login into the system. Doctor should securely login into the system. Hospital operator should securely login into the system. Hospital operator should securely login into the system .Database for calendar should be updated.

Post condition:

If successful, then daily schedule will be available to the Client, Doctor, Hospital Operator, System admin and new or updated schedule stored in the data base.

Main Success scenario:

1. Doctor/client/hospital operator/system admin successfully enters into the web site to login.
2. Doctor/client/hospital operator /system admin writes the user ID in the user ID field.
3. Doctor/client/hospital operator/system admin gives the password in the password field.
4. Doctor/client/hospital operator/system admin press OK button to login.
5. System match user ID given by Doctor/client/hospital operator/system admin with user ID stored in the database.
6. System match user password given by Doctor/client/hospital operator/system admin with user password stored in the database.
7. If user ID and password match with the user ID and password stored in the database then Doctor/client/hospital operator/system admin will be logged in securely.
8. Doctor/client/hospital operator clicks on the calendar field to access calendar.
9. Calendar page successfully load.
10. Doctor/client/hospital operator/system admin select specific date.
11. Doctor/client/hospital operator/system admin write specific time for that specific date.
12. Doctor/client/hospital operator/system admin see the calendar.
13. Doctor/client/hospital operator/system admin view the old schedule, update old schedule or set a new schedule.
14. System store updated old schedule or a new schedule.

Extension:

*a At any time system fails.

1. Doctor/client/hospital operator restarts the system.
- 2a. If Doctor/client/hospital/operator system admin doesn't write user ID in the user ID field then s/he will prompt to fill the filled again.
- 3a. If Doctor/client/hospital operator/operator system admin doesn't write user password in the user password field then s/he will prompt to fill the filled again.
- 7a. If user ID and password doesn't match with the user ID and password stored in the database then Doctor/client/hospital operator/operator system admin will not be logged in securely and prompt to fill the fields again.
- 9a. If Appointment schedule page doesn't load successfully then starts from step-8.

- 11a. If Doctor/client/hospital operator/system admin write specific time for that specific date then reminder can't send message to his/her mobile.
- 12a. Doctor/client/hospital operator/system admin see the calendar.
- 13a. Doctor/client/hospital operator/system admin view the old schedule, update old schedule or set a new schedule.
- 14a. If System doesn't store updated schedule or a new schedule then again start from step-10.

Process Browse Info:

Primary Actor:

Hospital Operator, Doctor, Client, System Admin.

Stakeholder and interest:

Hospital Operator, Doctor, Client, System Admin.

Doctor:

Doctor can get necessary information like another doctor's information, client information (how many clients are get appointed). Doctor also can maintain his/her schedule by browsing or accessing calendar. Doctors have the privilege to edit his/her profile in the editing page.

Hospital operator:

Hospital operator can add or delete doctors in a hospital using the hospital profile. Operator also can verify a doctor who requested to add in the hospital. Operator can do these works by browsing particular page.

Client:

Client can easily get appointment by using this site. Client can easily browse different specialized. Client can browse different hospital and can get required information from the hospital profile. Client can take proper decision to get better service by browsing doctors ranking page, hospital/chamber information page.

System Admin:

System admin can observe whether all the functionality of the website are running properly or not by browsing the required information.

Precondition:

The web site should be properly running.

Post condition:

After visiting the website the Stakeholders (Hospital Operator, Doctor, Client and System Admin) successfully save/edit/browse required information.

Main Success scenario:

1. Enter the site by valid site address.
2. Doctor login into his/her profile.
3. Edit Doctor's information by browsing required information.
4. Doctor's can see his/her clients by browsing doctor's client page.
5. Doctor's can get required information by browsing hospital/Doctor's page.
6. Client enters the website.
7. Client Search appropriate doctor by browsing the site.
8. Client matches his/her available time by the doctor's profile.
9. Client gives appointment to a doctor.
10. Doctor's appointed number will be increased.
11. Client can get necessary information like (doctor's phone number, location of a chamber or hospital) from the site.
12. Hospital operator enters the site.
13. Hospital operators login into a hospital.
14. Hospital operators add/delete doctors from the hospital profile.
15. A request to add the hospital into a doctor's profile will send the specific doctor.
16. System admin enter the site.
17. Check all the site pages to see whether the sites working properly or not

Extension:

*a At any time system fails.

1. Users restart the system.
 - 1a. If valid address not given then the website doesn't not load successfully. Enter valid address to load the website.
 - 2a. If Doctor does not fill the require fields correctly then he/she will prompt to give the valid login name and password again.
 - 4a. If the page doesn't not load successfully press F5 or Ctrl+F5.
 - 5a. If the page doesn't not load successfully press F5 or Ctrl+F5.
 - 6a. If valid address not given then the website doesn't not load successfully. Enter valid address to load the website.
 - 7a. If the page doesn't not load successfully press F5 or Ctrl+F5.
- 12a. If valid address not given then the website doesn't not load successfully. Enter valid address to load the website.
- 13a. If Hospital operator does not fill the require field correctly then he/she will prompt to give the valid login name and password again.
- 17a. If there is error on the web page then system admin will call the developers to fix the problem.

Process Verify

Primary Actor:

System administrator.

Stakeholder and interest:

Doctor:

System admin verify the valid client to get appointment from the doctor. Invalid users have no opportunity to get appointment after a certain time from a doctor which will decrease harassment of a doctor using the site.

Hospital operator:

Hospital operator is verified by the system admin. After verification the operator will get the full control to add/delete doctors in the hospital profile and can verify other doctors who are requested to add in the hospital.

Client:

System admin verify valid Hospital/chamber. So client can depend on the website to get valid doctor for appointment belongs to a valid hospital/chamber.

Precondition:

System administrator should have to go to the Bangladesh medical council/any government to collect approved hospital. Database should be updated.

Post condition:

If successful, Management gets overall report based on the system admin verification.

Main Success scenario:

1. System administrator goes to the Bangladesh medical council/any government Institution to collect approved hospital.
2. System administrator goes to the hospital to verify hospital operator.
3. If the hospital operator is valid and willing to take service from this site then System administrator will give him/her a user ID and password to register.
4. System administrator checks the client statistics to see the amount of users.
5. System administrator checks the clients who miss the appointment frequently.
6. If there is any fraud client then System administrator takes an effective action.
7. System administrator checks the client list those who have gone to the doctors chamber (given by doctor) with System administrator client booking list.
8. System administrator checks whether the doctor and hospital gives payment regularly or not.
9. System administrator checks the feedback comes from the client.

10. System administrator makes report based on the client's feedback to send to the management.

Extension:

*a At any time system fails.

1. System administrator restarts the system.

4a. If the hospital operator are not valid then cancel his/her registration.

4b. If the hospital operator are not willing to take the service from this site, don't give him/her ID and password to register.

8a. If the client list those who have gone to the doctors chamber (given by doctor) and System administrator client booking list doesn't match then send a report to the management

9a. If the doctors payment are not satisfactory then send a report to the management.

Process Manage System

Primary Actor:

System administrator.

Stakeholder and interest:

Doctor:

If the doctor forgot password then s/he gets necessary help from the system admin. Doctors request properly reach to the hospital operator so that doctor can register easily. Maintaining calendar properly helps doctor to do his/her work at scheduled time. Checking appointment helps doctor to collect client properly.

Client:

If the client forgot password then s/he gets necessary help from the system admin. Clients request properly reach to the system admin so that client can register easily. Maintaining calendar properly helps clients to do his/her work at scheduled time. Checking appointment helps client of getting appointment properly.

Hospital operator:

If Hospital operator forgot password then s/he gets necessary help from the system admin. Hospital operators request properly reach to the doctor so that Hospital operator can register easily. Maintaining calendar properly helps Hospital operator to do his/her work at scheduled time.

System administrator:

All problems handling come from client, doctor, hospital operator and making and sending reports to the management are done by system administrator. As a result customer and management satisfaction always goes to the system admin.

Precondition:

System administrator always be aware of handling problems and have to check system on due time.

Post condition:

If successful then all kinds of problem would be solved properly and making and sending reports are done appropriately.

Main Success scenario:

1. check validity of a user and if the user forgot password then set a new password and inform to that user
2. check records of a client
3. check if hospital's request reach to doctor
4. check if doctor's request reach to hospital
5. check if doctor's calendar is working properly
6. check if hospital's calendar is working properly
7. check if client's calendar is working properly
8. check if appointment is working properly
9. check if data backup is taken properly
10. Make report how much patient goes to which doctor
11. Make report how much patient goes to which hospital
12. Managing the advertisements.
13. Renew the server to make the site active.

Extension:

- 1a.If user can't verify then password will not be set and informed to that user
- 2a.If a client always takes appointment but don't keep it then that client should be notified and moderated
- 3a.If not reaches checks the system and ensures it
- 4a. If not reach check the system and ensure it
- 5a. If not working properly then fix it up
- 6a. If not working properly then fix it up
- 7a. If not working properly then fix it up
- 8a. If not working properly then fix it up
- 9a. If not taken then backup the data
- 10a. Ask for a percentage of the prescription to the doctor
- 11a. Ask for a percentage of the prescription to the hospital
- 12a. If the advertisement doesn't work properly inform the site developers.
- 13a. If doesn't Renew the server then site wont work and send a message to the management to renew the server.