

Design of UML Diagrams for WEBMED - Healthcare Service System Services

Dr.S.Suriya^{1,1}, Nivetha S¹

¹Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, India.

ABSTRACT

Healthcare service has huge demand these days as it really helps in managing a hospital or a medical office. The scope of Healthcare service systems is increasing by each day and it is true for the entire world. Some of these solutions include improved awareness about Healthcare services and health policies. The objective of this system is to provide medical assistance to people instantly with the help of technology. This system eradicates the cultural sensitivity that prevails in many hospitals and improvises the quality of medical assistance. The captivating features of this system are online doctor, medicines at doorstep, bulletin of awareness. The users can also navigate and choose among various insurance schemes that are displayed. Unified Modeling language (UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. It uses graphic notation to create visual models of software systems. This paper contains the UML diagrams for better understanding of the system with the help of Star UML tool. Use case diagrams are used during the analysis phase of a project to identify system functionalities. Class diagram represents the static view of an application. The class diagrams are the only UML diagrams, which can be mapped directly with object-oriented languages. Activity diagram is an important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity. The state machine diagram shows the different states of an entity and focuses more on how it responds to various events by changing from one state to another. Statechart diagram is used to capture the dynamic aspect of a system. State machine diagrams are used to represent the behavior of an application. The sequence diagram focuses on the messages that are passed during an interaction in a time based perspective. A Communication diagram models the interactions between objects or parts in terms of sequenced messages. It describes both the static structure and dynamic behavior of a system. Component diagrams are used to model the physical aspects of a system. It does not describe the functionality of the system but it describes the components used to make those functionalities. Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. UML is a modeling language used by software developers. UML can be used to develop diagrams and provide users with ready-to-use, expressive modeling examples. Some UML tools generate program language code from UML. UML can be used for modeling a system independent of a platform language. UML is a graphical language for visualizing, specifying, constructing, and documenting information about software-intensive systems. UML gives a standard way to write a system model, covering conceptual ideas.

Keywords: UML Diagrams, health service systems, usecases, activity, statechart, sequence, communication, classes, deployment, component

Received on 25 May 2020, accepted on 20 June 2021, published on 01 February 2023

Copyright © 2023 Dr.S.Suriya *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetel.v8i1.3015

1. INTRODUCTION

Problem Statement: The primary objectives of Webmed healthcare system are to enable all citizens to receive health

care services whenever needed, and to deliver health services that are cost-effective and meet pre-established standards of quality. The main functions of this system deals with finance, health A-Z, resources, drugs and supplements, news and experts, payment and feedback. Register function allows the patients or the care givers to register into the website. Login

¹ Corresponding author. Email: suriyas84@gmail.com

function allows the patients to access the website. Financing focuses on purchase of insurance. Health A-Z displays all the diseases along with their symptoms. Resources function consists of the sub functions including symptoms checker, health calculator, find a doctor based on the geographical location of the patient, insurance guide, and ambulance providence. Drugs and supplements include online medicine delivery, where people could shop for medicines online. News and experts function is to provide health awareness and threats that are prevailing. This function also gives information regarding counselling programs and blood donation camps. Payment function is to reimburse providers for services delivered. Feedback function collects user reviews for the website.

Major functionalities and its description:The major functionalities of this system along with their description are as follows: Register -this functionality acts as a membership functionality that allows the users, patients or care-givers to register into this website inorder to access all the resources. Login - this functionality authenticates the user to provide access permissions. Facilities - this functionality allows us to access all the options available in the website. Logout -this functionality ends the access to the website.Finance -this functionality allows to purchase insurance, or to pay for health care services consumed. Health A-Z - this functionality provides details of all the diseases.Resources - this functionality has sub-functions namely symptoms checker, health calculator, find a doctor, insurance guide and ambulance providence. Drugs and supplements - this functionality provides online delivery of medicines. This

functionality takes the doctor’s statement for the issue of medicine and the patient’s willingness to buy it online.News and experts - this functionality provides health awareness and threats that are prevailing. Payment - this functionality deals with the payment for the services consumed. This functionality is split into two sub-functions namely, payment for doctors, for the services issued and payment for online med- delivery.Feedback - this functionality collects user reviews for this website.

System architecture: System architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system. This architecture diagram shown in Figure1 includes all the functionalities of the system in an ordered manner based on the sequence of occurrence of these functionalities. It represents the complete data flow starting from the register functionality through which the user could access the system to the logout functionality that ends the access of the user to the system. Initially, the user must register onto the system. Once registered, the user would be able to log on to the system and could access all the services offered – finance, news and experts, resources that includes find a doctor, health A-Z that includes health calculator and fitness calculator, drug and supplements, payment and feedback. The payment is done through the insurance for the services consumed. After consuming the services, the user could log out of the system

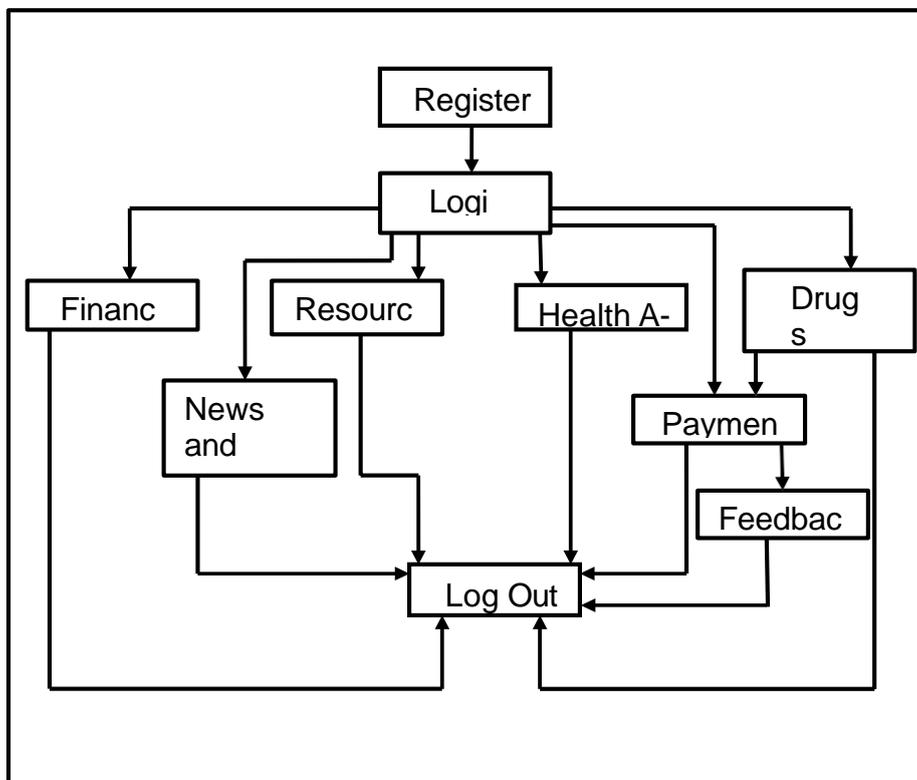


Figure 1. System Architecture

1.1 Data Flow Diagrams:

A Data Flow Diagram (DFD) is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow; there are no decision rules and no loops. A data flow diagram can dive into progressively more detail by using levels and layers. DFD levels are numbered 0, 1 and 2.

DFD Level 0 is also called a Context Diagram. It is a basic overview of the whole system or process being analyzed or modelled. It is designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. This level 0 DFD diagram as shown in figure 2 consists of the healthcare service provider system – WEBMED and the actors who affect those systems – the users that is the patients and the service providers who are responsible for the maintenance of the system. This is the basic diagram that could be easily understood by a wide audience. DFD Level 1 as shown in figure 3 provides a more detailed breakout of pieces of the Context Level Diagram. We will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub processes

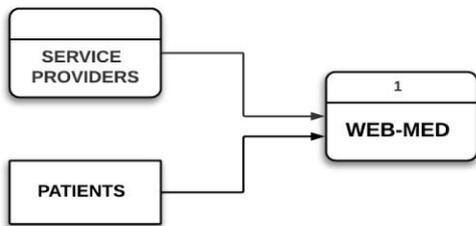


Figure 2. Level 0 DFD

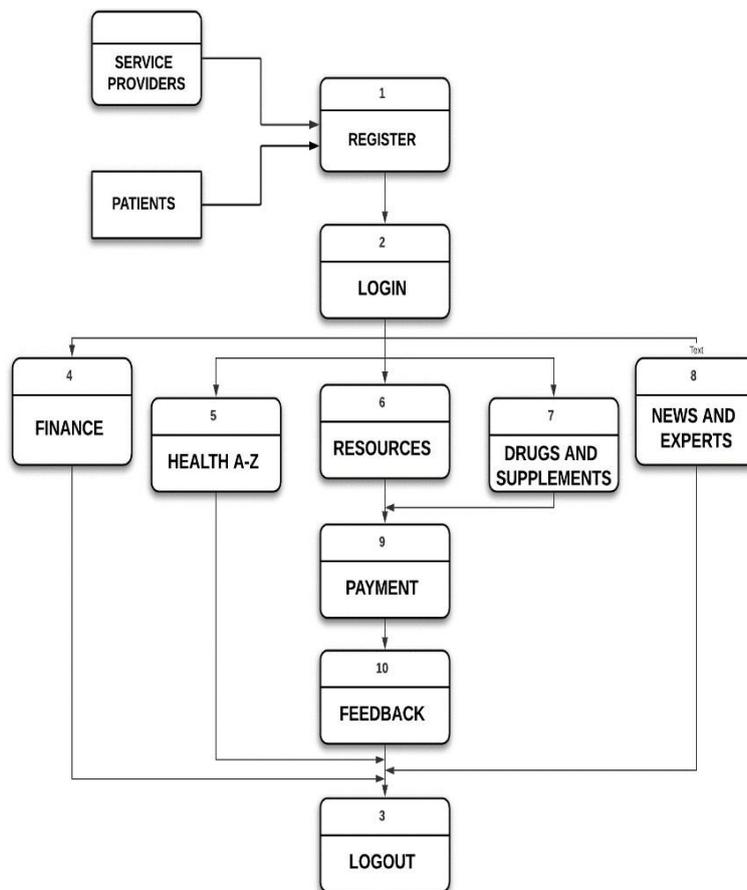


Figure 3. Level 1 DFD

These systems too have the same external entities – patients who require the service and the service providers who offer those services via the system. The webmed system is broken down into a number of sub processes. These sub processes represents the functionalities of the system – register, login, finance, health A-Z, resources,

drugs and supplements, news and experts, payment, feedback and logout. Payment is done for find a doctor and the drugs and supplements services provided. The user could also give the feedback for the services available. The table1 describes comparison between the existing system and proposed system

Table1. Comparison of Proposed System and Existing System

Existing system	Additional features of Proposed system with reference to existing system
Register	Same as that of existing system
Login	
Logout	
Finance	
Health A-Z	It acts as a dictionary of diseases along with their symptoms.
Resources	Same as that of existing system
Drugs and supplements	This functionality provides online delivery of medicines.
News and experts	This functionality provides health awareness and threats that are prevailing.
Payment	Same as that of existing system
Feedback	This functionality collects user reviews for this website.

2. UML DIAGRAMS:

2.1 UML USECASE DIAGRAM

Use cases are used during the analysis phase of a project to identify system functionality. They separate the system into actors and use cases. Actors represent roles that are played by users of the system. Users may be humans, other computers or even other software systems. The major components of UML usecase diagram are the actor – to specify the actor, usecase – to represent all the functionalities of the system, system boundary – to specify the scope of the system, and dependency – to represent an existence relationship. It contains the association and directed association to represent an interaction and a strongly influenced interaction between the actor and the usecase respectively. Figure 4 shows the UML Use case diagram of the proposed system.

2.2 UML CLASS DIAGRAM

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. The major component of class diagram is the class. A class consists of three compartments namely the class name, the attributes and the operations. The interaction between two classes are done via relationships. There are abstract classes that represents a list of generalized operations where the subclasses are free from duplicating those operations. It consists of interfaces that represents a collection of operations that is used to specify services to other classes or other components. Abstract class represents the common attributes and operations that are connected using an association relationship. There is another class called template class that represents a family of classes that have a structure and behaviour independent of the formal parameters. Figure 5 shows the UML class diagram of the proposed system.

2.3 UML ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. The major components of activity diagram are action, that represents the basic unit for behaviour, horizontal or vertical swimlanes to represent sequential or parallel activities, start state that represents the starting point of the system, final state that represents various finishing points of the system and flow final that represents termination of any path. Activity diagrams deal with all type of flow control by using different elements such as fork, join, decision and merge. Figure 6,7 shows the UML Activity diagram of the proposed system.

2.4 UML STATEMACHINE DIAGRAM

State machine diagrams are also called as state chart diagrams. State machine diagrams are used to capture the behaviour of a software system. UML State machine diagrams can be used to model the behaviour of a class, a subsystem, a package, or even an entire system. Statechart diagrams provide us an efficient way to model the interactions or communication that occurs within the external entities and a system. A state of an object is controlled with the help of an event. Statechart diagrams are used to describe various states of an entity within the application system. The major component of this UML diagram is the state that represents behaviour of an entity that contains the entry action, do action and the exit action. This diagram also includes those components that are in activity diagram like initial state, final state and flow final. The junction and choice points are used instead of join and fork. This diagram also has transition that represents movement from one state to another. It also includes the submachine state which is the collection of disjoint or concurrent substates. Figure 8 shows the UML State Machine diagram of the proposed system.

2.5 UML SEQUENCE DIAGRAM

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when. The major component of the sequence diagram is the object that represents every individual entity of the system. It forms the elements in the horizontal axis of sequence diagram. The other component is the lifeline that forms the vertical axis of this diagram representing the lifetime of the object. The other important component is the focus of control –

‘activation bar’ representing active state of the object. Messages are used to represent the transaction between the objects. There are 8 types of such messages – synchronous, reply, asynchronous, self, lost, found, create and destroy message. This diagram also includes fragments that represent any kind of conditional checking and looping messages. There are 7 types of fragments – PAR, ALT, OPT, NEG, LOOP, SD and REF. Figure 9 shows the UML Sequence diagram of the proposed system. Figure 9 shows the UML Sequence diagram of the proposed system.

2.6 UML COLLABORATION DIAGRAM

UML communication diagrams, like the sequence diagrams is a kind of interaction diagram that shows how objects interact. A communication diagram is an extension of object diagram that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, communication diagram shows the messages the objects send each other. The communication diagram represents the collaboration of objects in order to portrait the flow of activities that takes place in a particular usecase. It helps to frame a systematic procedure for explaining the usecases. The major component of collaboration diagram is the object that represents every individual entity of the system along with its lifeline which is represented as a line below the name of the object. Link is used to represent the communication between the objects. Any finite count of messages can be passed over a single link. The collaboration diagram has no fragments. Hence all the conditional and looping statements are represented using messages. There are 10 messages in this synchronous, reply, asynchronous, self, create, destroy, iterative, conditions, iterative conditional and mutually exclusive messages. Figure 10 shows the UML Collaboration diagram of the proposed system.

2.7 UML COMPONENT DIAGRAM

When modeling large object-oriented systems, it is necessary to break down the system into manageable subsystems. UML component diagrams are used for modeling large systems into smaller subsystems which can be easily managed. The component diagram is used to cross-verify whether all system functionalities are covered in planned development. It gives importance for interaction between various parts of the system through a set of well-defined interfaces. The major component of the component diagram is the component that represents every individual module of the system which encompasses all its contents and found to be reusable in nature. The other component is the stereotypes that represents generic module of a system. There are three stereotypes – subsystem that represents a particular module that can stand alone, service that represents either internal service offered to external environment or vice versa, process that represents those

services that are internally offered by the system. The component is the interfaces – required, provided and assembly interfaces. The artifact that refers to any document or file that is involved during the processing is yet another component. A part that is either an attribute or a operation. A port enables communication between a classifier and an external environment. Figure 11 shows the UML Component diagram of the proposed system.

2.8 UML DEPLOYMENT DIAGRAM

The deployment diagram maps the software architecture created in design to the physical system architecture that

executes it. In distributed systems, it models the distribution of the software across the physical nodes. The deployment diagram gives much importance for runtime processing elements that enables execution of software modules. The major component of the deployment diagram is the node that refers to any component that has both memory power and processing capability. The device node that refers to the runtime processing element. This diagram also includes the execution environment node. This diagram also includes the port, part, artifact, the dependency. Figure 12 shows the UML Deployment diagram of the proposed system. The Table2 summarizes functionalities of all discussed UML Diagrams

Table 2 Functionalities of UML Diagrams

Name of the UML diagram	Purpose
UML Usecase diagram	To represent functional requirements of a system under consideration.
UML Class diagram	To represent characteristics and behaviour of a system.
UML Activity diagram	To represent flow of control in a system.
UML State machine diagram	To represent the condition of a system or a part of the system at finite instance of time.
UML Sequence diagram	To represent the ranking or ordering of messages during an interaction between the entities of the system.
UML Collaboration diagram (or) UML Communication diagram	To represent the interaction between objects involved to perform the behaviour of an usecase.
UML Component diagram	It focusses more on design elements that require or provide interfaces in order to interact with various constructs of the system.
UML Deployment diagram	It focusses on topology of physical components over which the software components are deployed.

3. Proposed Design - UML Diagrams

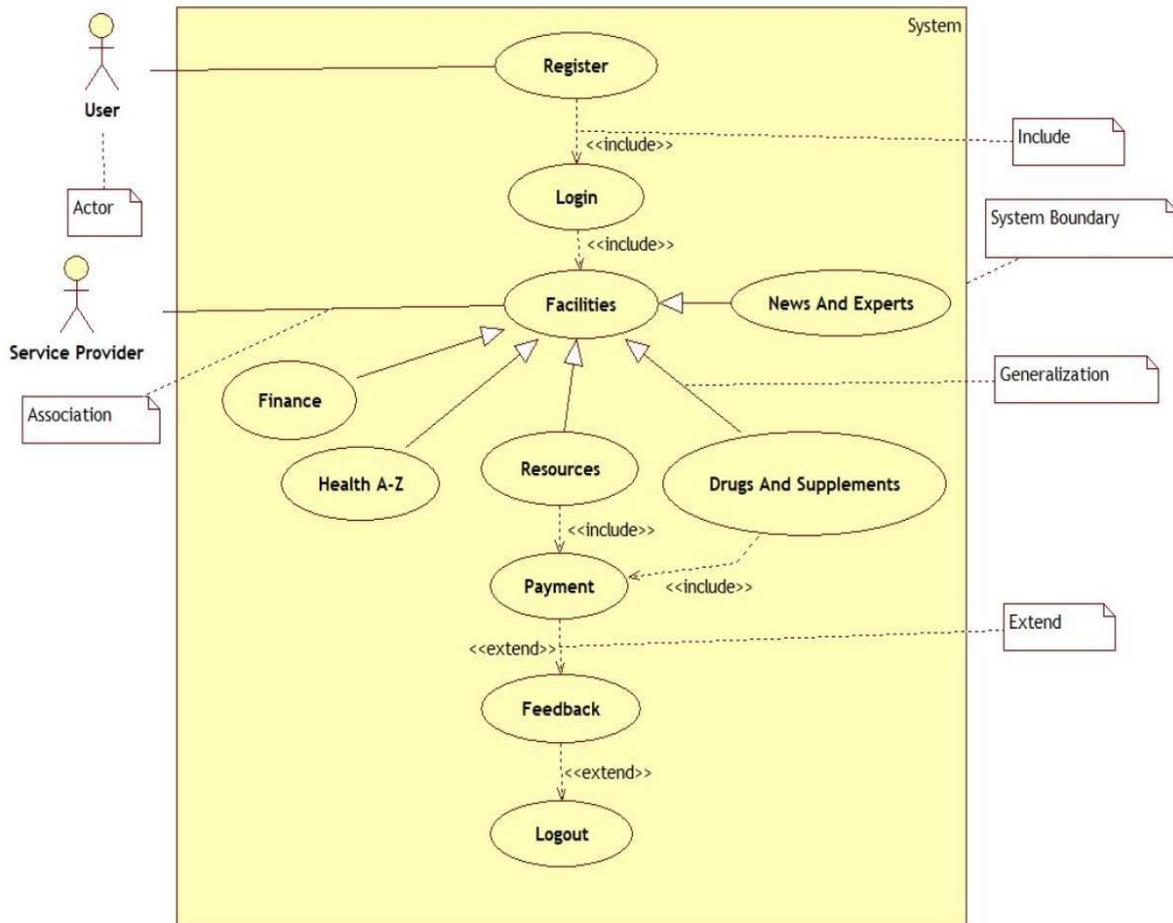


Figure 4. UML USECASE DIAGRAM

Usecase Specification for the UML Usecase Diagram generated using Star UML Tool

3.1 ACTORSPECIFICATION

PATIENT

Description

This actor plays the role of a patient who can register into the website and can make use of all the services provided by the website. This actor can also play a role of a care giver.

SERVICE PROVIDER

Description

This actor acts as the master of this website who provide certain services. This actor maintains a database of doctors

which helps a patient to find a doctor and looks after the finance, insurance and payment requirements. He also gives descriptions regarding various diseases which can be referred by the users. He also provides facility for the users to purchase drugs and other medicines online and spreads awareness through news regarding the prevailing threats.

USE-CASE SPECIFICATION

1. DRUGSANDSUPPLEMENTS

Description

This functionality provides online delivery of medicines.

1.1 Flow of Events

- **Basic Flow:** It allows the users to search for and purchase drugs and other medical supplies like syringes, bandages and dressing, earloop face masks etc.

- **Alternative Flow:** Alternate flow for this usecase is : HEALTH A-Z, RESOURCES,FINANCE,NEWS AND EXPERTS.

1.2 Pre-Conditions

- The user must have logged on to the website in order to make use of this functionality.

1.3 Post-Conditions

- The drugs and medicines database gets updated accordingly.

2. FACILITIES

Description

The service provider controls all the facilities in the website through this usecase and the user can utilize these facilities.

2.1 Flow of Events

- **Basic Flow:** The user can obtain the services like:HEALTH A-Z,RESOURCES,DRUGS AND SUPPLEMENTS,NEWS AND EXPERTS and FINANCE.
- **Alternative Flow:** The user can provide his feedback for improvement of the website.

2.2 Pre-Conditions

- The user must have logged on to the website in order to make use of this functionality.

2.3 Post-Conditions

- By getting into this usecase,the user can make use of the available services.

3. FEEDBACK

Description

This Feedback section helps the users to give their honest opinions about the quality of services that has been provided.It also helps them to give their suggestions for further improvements.

3.1 Flow of Events

- **Basic Flow:** It verifies whether the feedback is given or not.
- **Alternative Flow:** LOGOUT

3.2 Pre-Conditions

- The payment function should have been completed.

3.3 Post-Conditions

- The feedback form is submitted

4. FINANCE

Description

Negotiations regarding the financial status are done ,eligibility for the insurance is checked and the appropriate schemes are issued.

4.1 Flow of Events

- **Basic Flow:** checks if the user has availed any insurance for his health if not it Checks for insurance availability.
- **Alternative Flow:** Instead of this usecase, the following usecases can be executed : HEALTH A-Z,RESOURCES,DRUGS AND SUPPLEMENTS, NEWS AND EXPERTS.

4.2 Pre-Conditions

- The patients should have a valid bank account.

4.3 Post-Conditions

- The patient would have insured his life.

5. HEALTH A-Z

Description

This functionality acts as a dictionary of the diseases along with their symptoms and the drugs that can be consumed.

5.1 Flow of Events

- **Basic Flow:** It provides catalog of all the diseases with its symptoms.
- **Alternative Flow:** Instead of this usecase, the following usecases can be executed : FINANCE,RESOURCES,DRUGSAND SUPPLEMENTS, NEWS AND EXPERTS

5.2 Pre-Conditions

- The user should have logged onto the website.

5.3 Post-Conditions

- The user is now aware of the drugs that must be consumed for a specific disease.

6. LOGIN

Description

This functionality allows the patients and care givers to login into the website to get the access of all the resources.

6.1 Flow of Events

- **Basic Flow:** It allows the user to access all the resources that are available in the website.
- **Alternative Flow:** After this usecase is executed,then the control flows through any one of the following usecases :FINANCE, HEALTH A-Z, RESOURCES,

DRUGS AND SUPPLEMENTS ,NEWS AND EXPERTS.

6.2 Pre-Conditions

- The user should have registered into the website

6.3 Post-Conditions

- The user is given the complete access to all the resources available in the website.

7. LOGOUT

Description

This functionality ends the access to the website. So the user must login again to access all the resources.

Flow of Events

- **Basic Flow:** It ends the access of the resources by the user.
- **Alternative Flow:** It has no alternate flow.

Pre-Conditions

- The user should have logged onto the website.

Post-Conditions

- Once this usecase is executed ,the user is denied from the resources provided by this website.

8. NEWSANDEXPERTS

Description

A bulletin of News regarding the latest diseases that prevails in the Environment,Expert opinions to avoid those communicable and non communicable diseases, Awareness about the diseases will be displayed publicly.

8.1 Flow of Events

- **Basic Flow:** The current affairs about the diseases are well received by the users.
- **Alternative Flow:** Instead of this usecase, the following usecases can be executed : HEALTH A-Z,RESOURCES,DRUGS AND SUPPLEMENTS,FINANCE.

8.2 Pre-Conditions

- The user should have logged in.

8.3 Post-Conditions

- The user is enlightened about the prevalent diseases.

9. PAYMENT

Description

This functionality deals with the payment for the services consumed.

9.1 Flow of Events

- **Basic Flow:** This functionality involves the payment for the services provided by the doctor.
- **Alternative Flow:** After executing this usecase,the control moves to:FEEDBACK

9.2 Pre-Conditions

- The user should have consumed the services provided by the doctor or should have purchased medicines online.

9.3 Post-Conditions

- After executing this usecase, the user has made the payment for the medicines purchased or the service providers have made the payment to the doctors.

10. REGISTER

Description

This acts as a membership functionality that allows the patients or care-givers to register into this website in order to access all the resources

10.1 Flow of Events

- **Basic Flow:** It allows the user to register into the website.
- **Alternative Flow:** After the register usecase is executed, the usecase that can be executed is:LOGIN functionality.

10.2 Pre-Conditions

- This is no precondition as register is the first usecase to be executed.

10.3 Post-Conditions

- After the register usecase is executed,the user will be given a webmed membership.

11. RESOURCES

Description

The resources such as health calculator, ambulance services, symptoms checker,find a doctor, insurance guidance are unveiled in this segment.

11.1 Flow of Events

- **Basic Flow:** Thus the resources are gathered by the users.

- **Alternative Flow:** Instead of this usecase, the usecases that can be executed are: HEALTH A-Z, NEWS AND EXPERTS,DRUGSAND SUPPLEMENTS,FINANCE.

11.3 Post-Conditions

- The users have enjoyed the privileges of accessing the resources

11.2 Pre-Conditions

- The availability of the resources should be ensured.

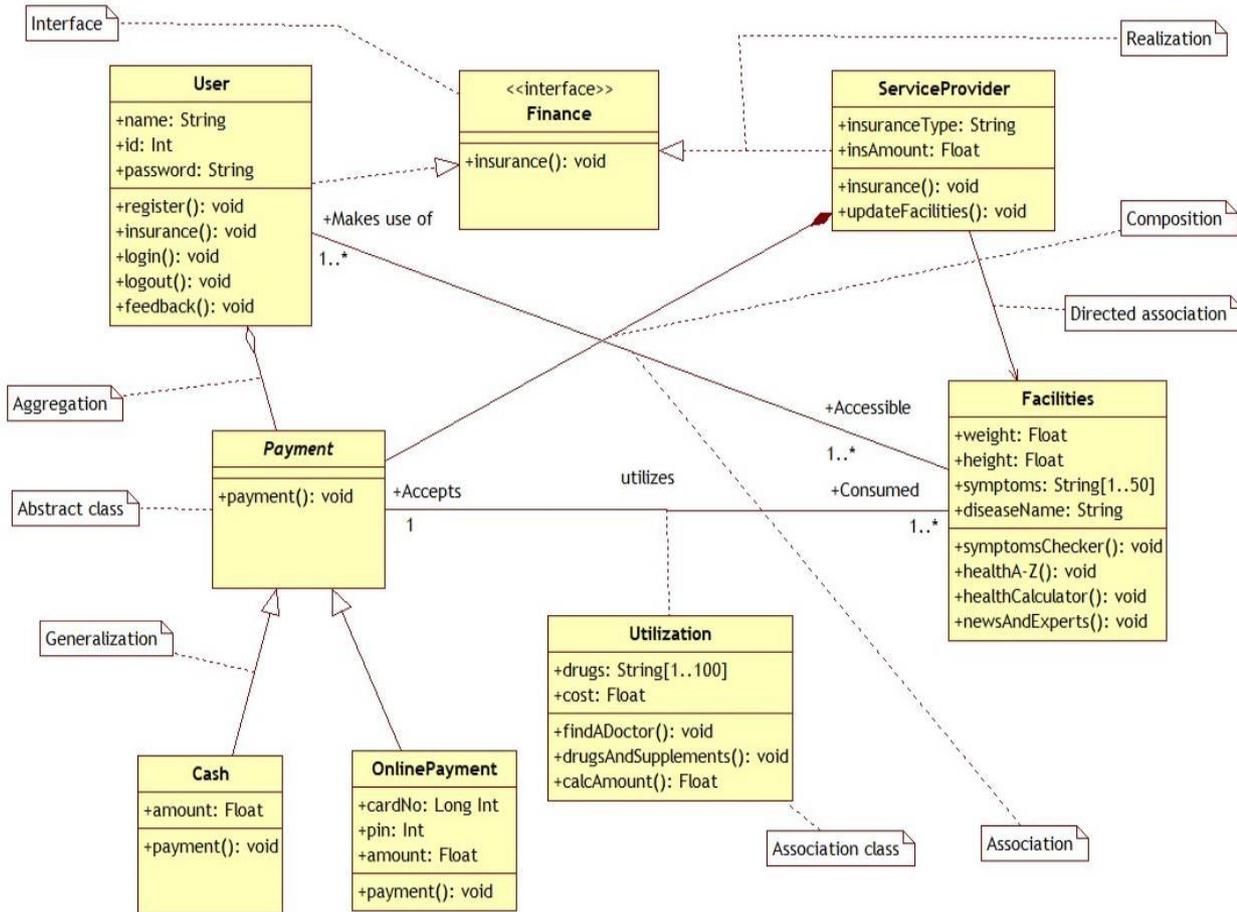


Figure 5. UML CLASS DIAGRAM

The two actors, user and the service providers can access the resources in the website. The facilities class consists of the functions like News and Experts, Health A-Z and Health Calculator. The finance class checks if the user has health insurance and if not alerts the service provider who buys the insurance for the user. Realization relationship is used to link the user and the service provider with the finance class. The payment class involves the payment by the user for the services consumed. The payment can either be an online payment or through cash on delivery of drugs and other supplements. Thus the generalization relationship is used to represent the cash and onlinepayment of the payment class. The payment class is loosely related to the user class and it is strongly related to the service provider class as the

service provider takes care of all the payment procedure. The resources like find a doctor, that suggests a suitable doctor specialized for a disease and purchase of drugs and supplements are the resources for which the payment is done. So these resources are common to both the facilities class and the payment class. Hence these resources are kept in an association class named as 'utilization'.

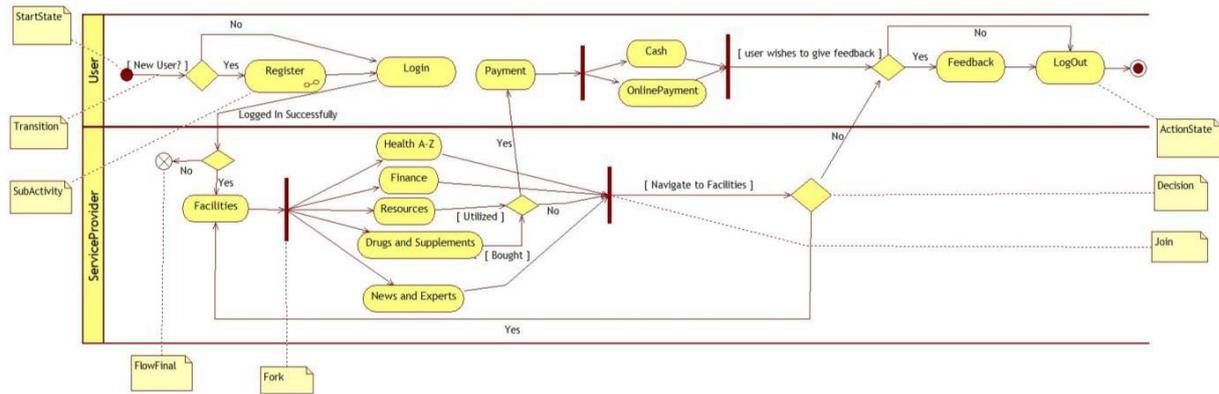


Figure 6. UML ACTIVITY DIAGRAM(1)

The activity diagram represents the flow of data through the system. It represents the entire flow of data from the registration of the user, followed by logging on to the website with username and password. Once the user has logged on to the system, the user could access all the

resources and can navigate through the resources again until the user wishes to log out the system. The user needs to pay for the services which they acquired. The user could also give the feedback for the system if the user opts it.

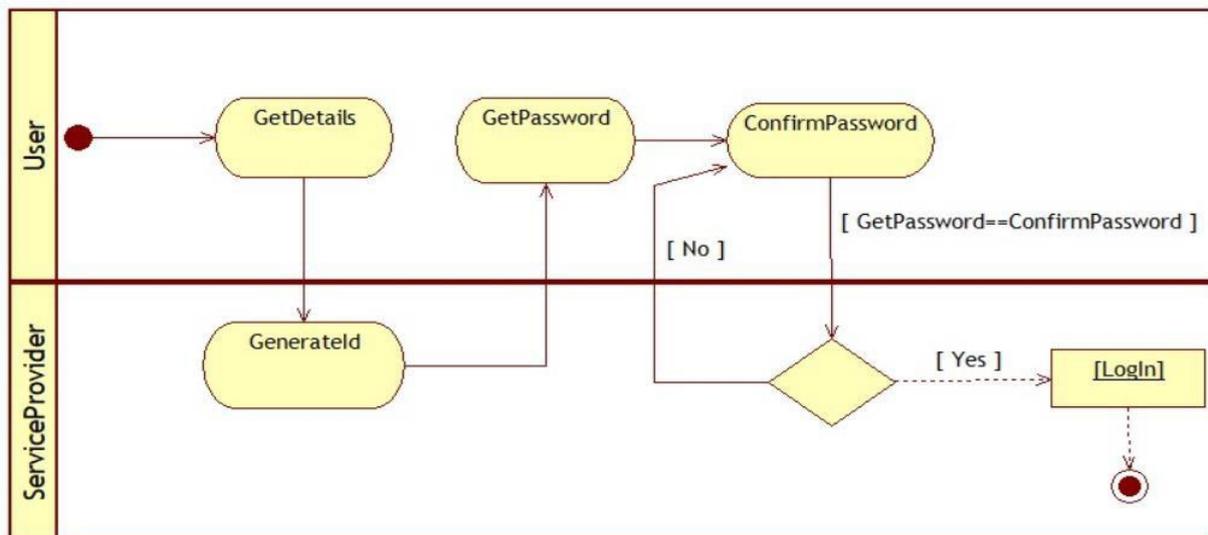


Figure 7. UML ACTIVITY DIAGRAM (2)

In this activity diagram, the register is a sub activity. The register has an individual flow of control that enables the user to enroll them into the system. The activity diagram above illustrates the flow that takes place in the register function. The user is prompted to give all the details. The service provider then generates the random id and receives the password from the user and confirm password. If the password and confirm password are the same

then the user is registered to the system else the user is prompted to give the text for the confirm password field alone.

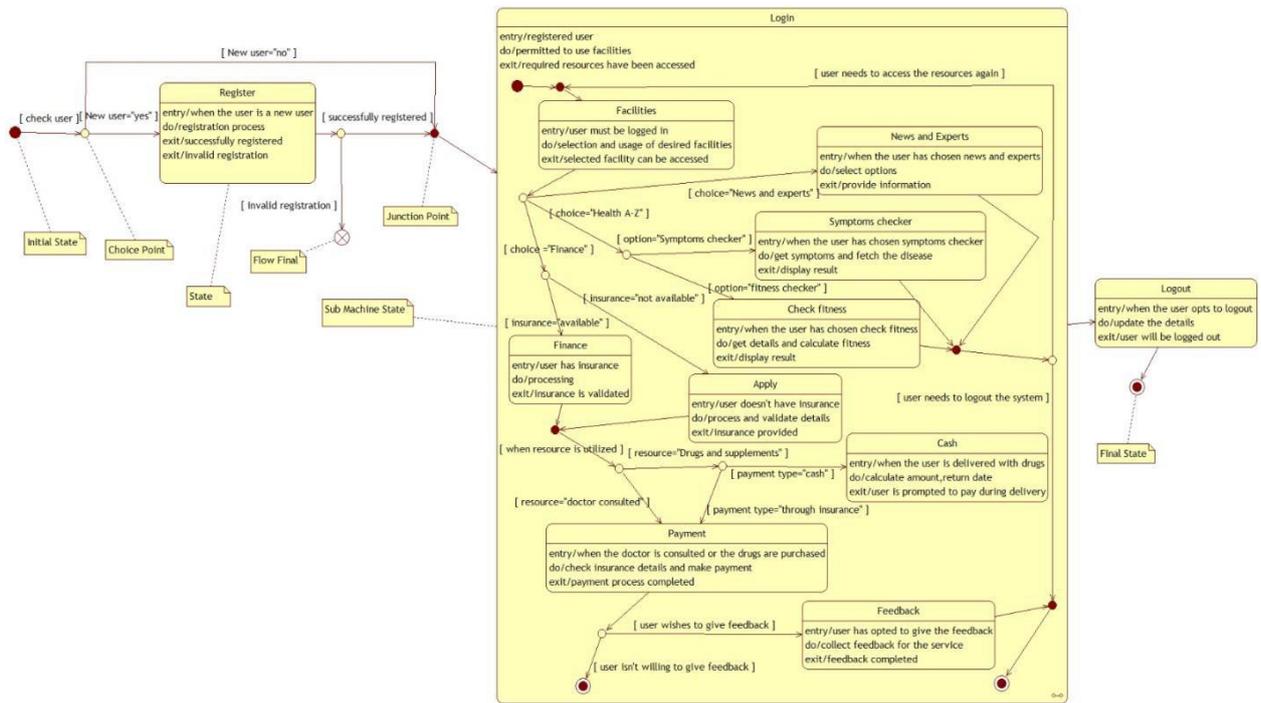
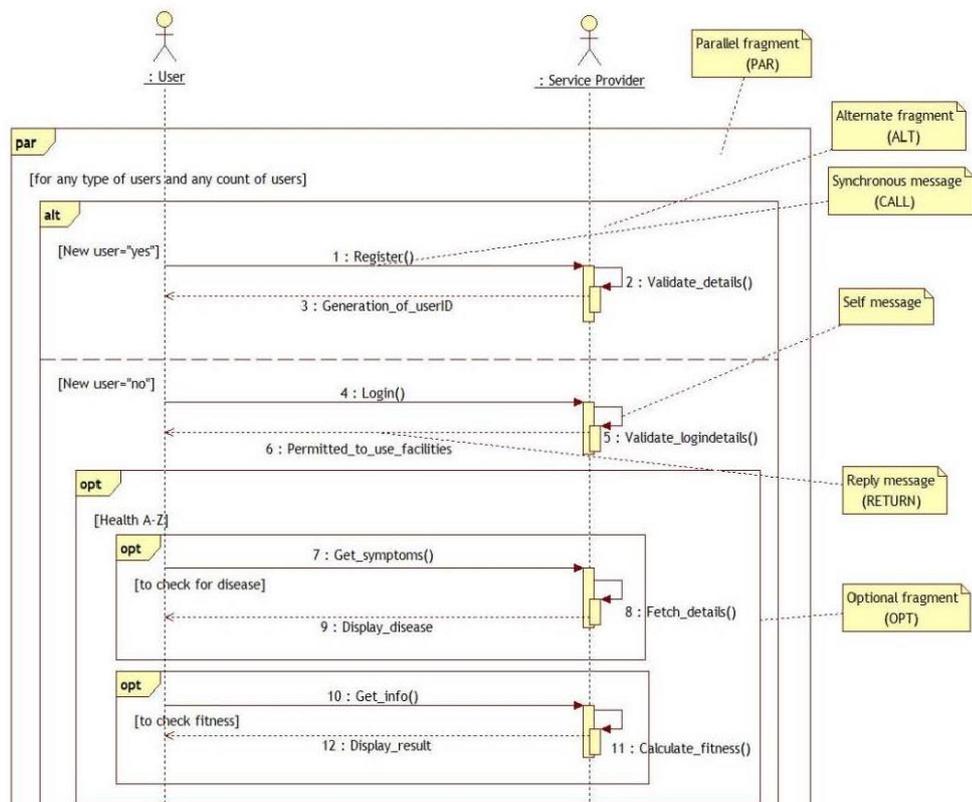


Figure 8. UML STATE MACHINE DIAGRAM



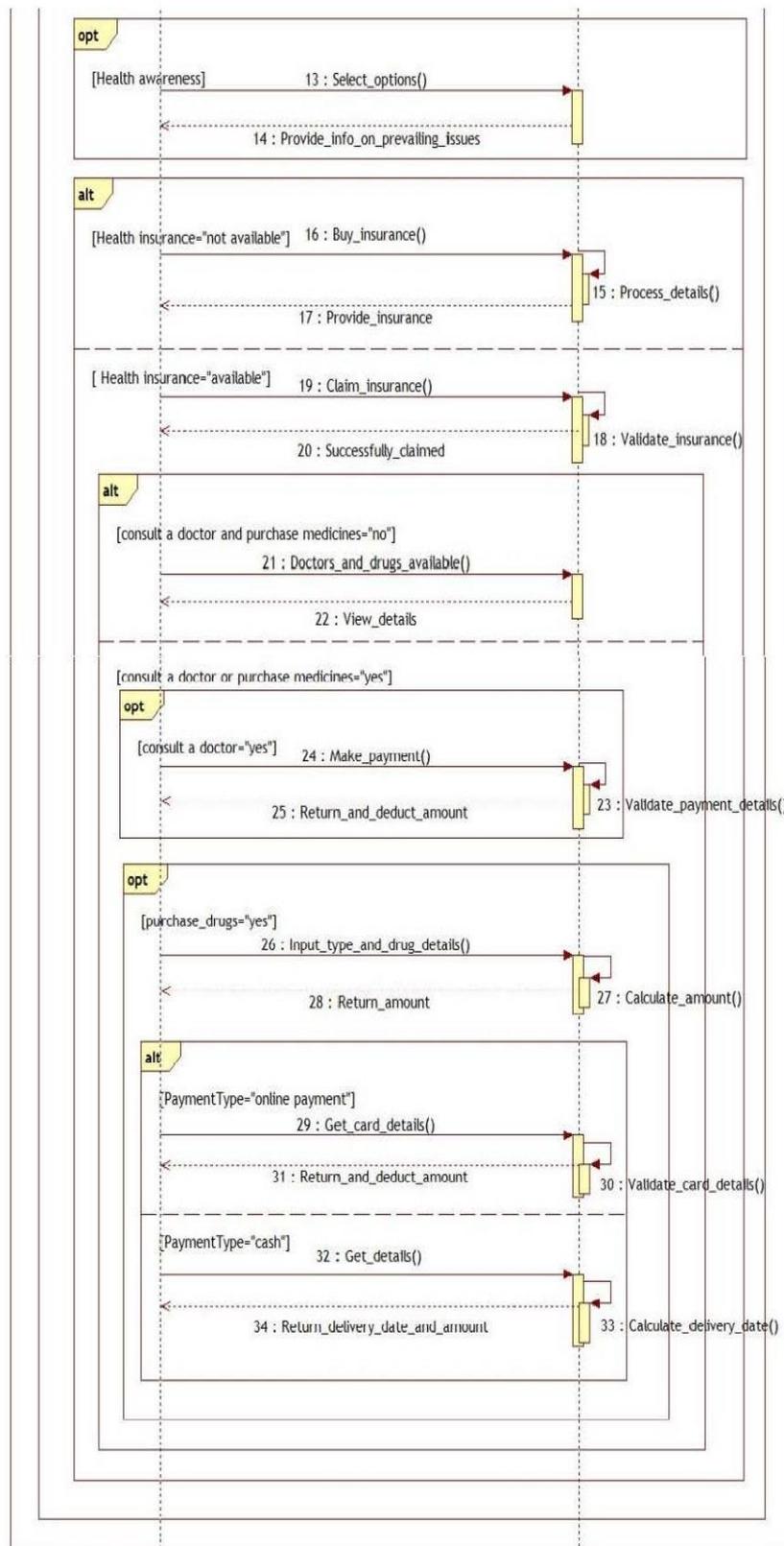


Figure 9. UML SEQUENCE DIAGRAM

The system first checks if the user is a registered user, if yes the user is prompted to the login state, else the user is asked to register into the system and is prompted to the register state. Once logged into the system, the user could access the facilities based on the choice of the user. Based on the choice of the user, the user is prompted to the required state. In this diagram, the login is a submachine state that consists of many disjoint states.If the user needs to find a doctor or needs to purchase drugs then the user is checked for the availability of insurance. If not the system buys an insurance for the user and is allowed to use the services.The payment is done for the services consumed using the payment state. Once the user has accessed the resources, the user could again loop on all the resources again if required, else the user could log out of the system.Before logging out of the system, the user could also give the feedback for the services provided by the system if the user wishes to do so.

The interaction between the user and the service provider is clearly depicted in this diagram. The user is checked if the user is already registered to the system or not. If not registered to the system, the register function takes place. If registered the user can logged in the system with username and password. For this if else case, the alternate fragment is used.After logging on to the system, the user is correctly prompted to the functionality based on the choice of the user. Hence to select the resource, the optional fragment is used.In order to access the find a doctor and the drugs and supplements, the user is checked for the insurance. If not, the service provider buys insurance for the user. This insurance is used for the payment of the resources utilized.The request from the user is sent as a synchronous message as the user needs the response for the request. The response is sent as a reply message. The complete flow of messages from the register to the payment and delivery of the purchased drugs is clearly depicted in the sequence diagram.

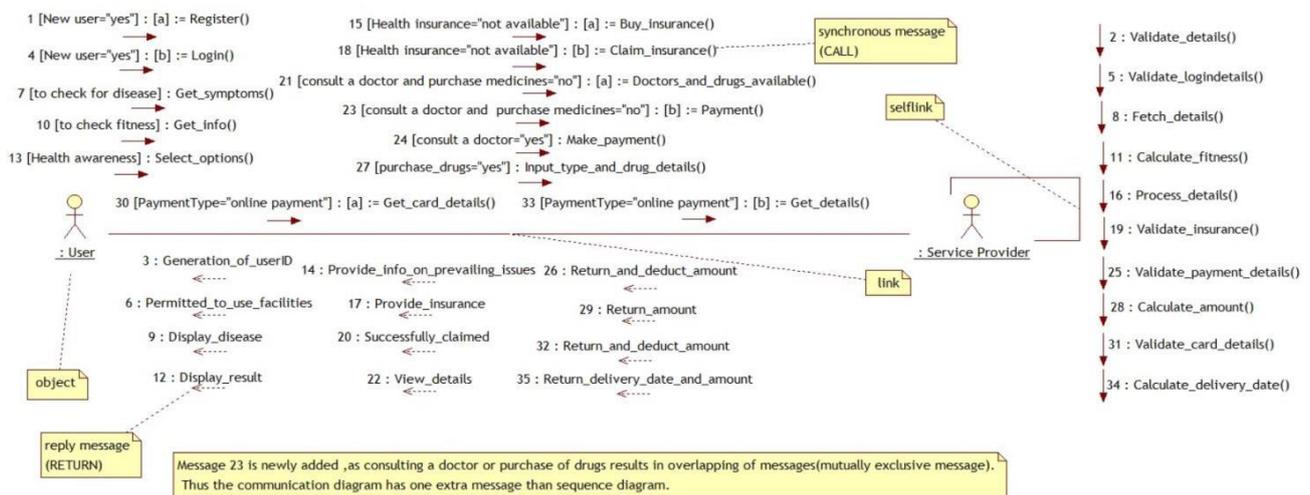


Figure 10. UML COMMUNICATION DIAGRAM

It focuses on the interaction between the actors in the system same as that of the sequence diagram.The collaboration diagram does not consist of any fragments. Hence the messages are used to represent the conditional and looping statement. The sequence diagram has 34 messages. But the collaboration diagram has 35 messages.

This extra one message (23) is used in order to prevent the overlapping of messages. Since the entire payment function involves overlapping of messages, an extra message is used to indicate the start of the payment function.

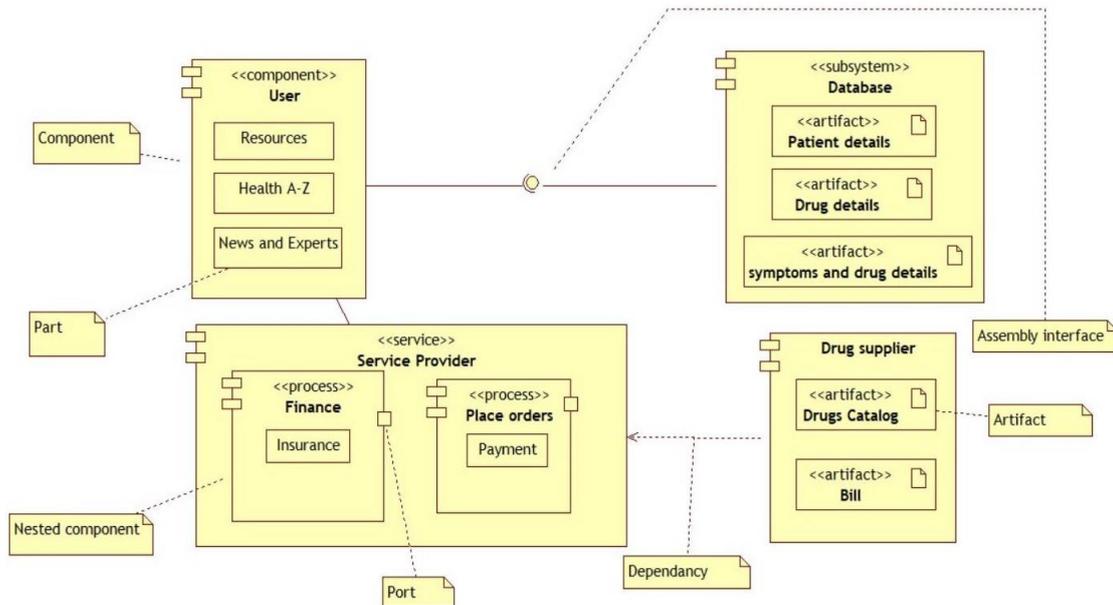


Figure 11. UML COMPONENT DIAGRAM

A component is a replaceable and executable piece of a system whose implementation details are hidden. A component provides the set of interfaces that a component realizes or implements. Components also require interfaces to carry out a function. Components also require interfaces to carry out a function. UML Component diagrams are used to represent different components of a system. In this diagram, every component represents an individual module which encompasses all its attributes and operations. The

user component can access all the resources. The user component consists of the resources, health A-Z, news and experts as it can access those directly. The database subsystem consists of all the artifacts that are required for the system. The service provider is a service that takes care of the processes like financing, place order to buy those drugs that are out of stock. Drug supplier component supplies the drugs to the service providers for which the payment is done.

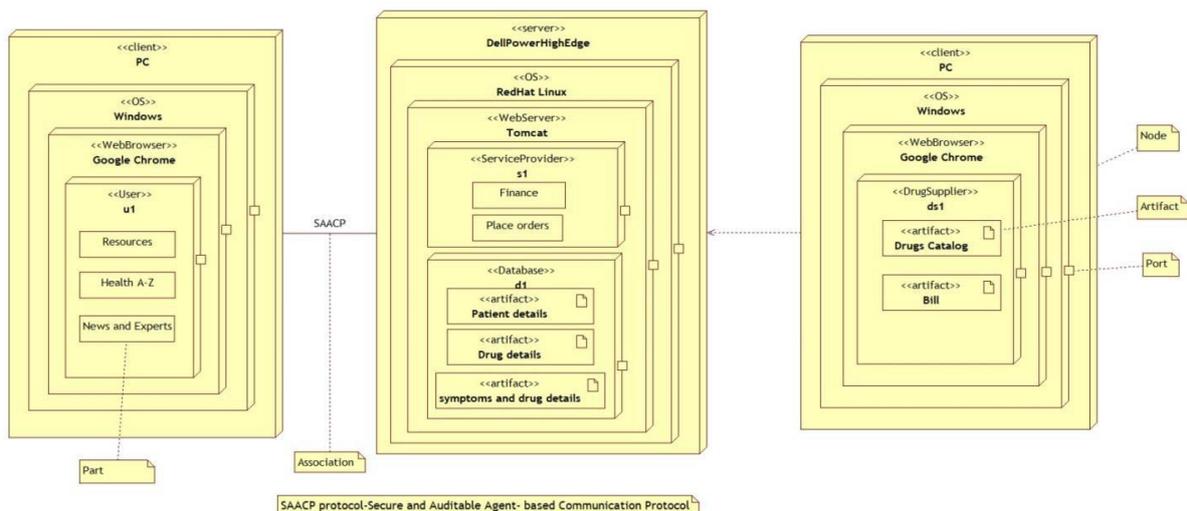


Figure 12. UML DEPLOYMENT DIAGRAM

The nodes and the Execution Environment Node are used to represent the OS, database and the web browser. The attributes and the operations are represented using parts as that of the component diagram. The database is maintained in the service provider as it needs to search for the details of the patient in the event of register, login, finding a doctor and payment. The drug supplier node is used to illustrate the attributes and operations involved in representing the ordering and purchasing of the drugs. The protocol that is used between the user and the service provider is SAACP. SAACP stands for Secure and Auditable Agent-based Communication Protocol which is often used by most of the hospitals to maintain the details of the patients in a secure way. Each node has parts that denote the attributes and operations that is can involve.

4. CONCLUSIONS AND DISCUSSIONS

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML diagrams helped to trace out a system for the given problem statement. The UML diagrams provided a step by step modeling of the system with each diagram having its own unique purpose. UML helps the project teams to communicate, explore potential designs, and validate the architectural design of the software. UML also enables us to automatically generate the code for some of its diagrams. This system can be enhanced with some special functionalities like ambulance providence through which the control unit could check for the ambulance near the patient and provide service. The other functionality is blood donation through blood donation camps, users are given the information about when and where these camps are set. These functionalities could be easily included using the UML diagrams.

REFERENCES

- [1] Milovancevic, M., Marinović, J. S., Nikolić, J., Kitić, A., Shariati, M., Trung, N. T., ... & Khorami, M. (2019). UML diagrams for dynamical monitoring of rail vehicles. *Physica A: Statistical Mechanics and its Applications*, 121169.
- [2] Cvetković, J., & Cvetković, M. (2019). Evaluation of UML diagrams for test cases generation: Case study on depression of internet addiction. *Physica A: Statistical Mechanics and its Applications*, 525, 1351-1359.
- [3] Hashmani, M. A., Zaffar, M., & Ejaz, R. (2019). Scenario Based Test Case Generation Using Activity Diagram and Action Semantics. In *Human Factors in Global Software Engineering* (pp. 297-321). IGI Global.
- [4] Mala, D. J. (2019). IoT Functional Testing Using UML Use Case Diagrams: IoT in Testing. In *Integrating the Internet of Things Into Software Engineering Practices* (pp. 125-145). IGI Global.
- [5] Voit, N., Kanev, D., Kirillov, S., & Ukhanova, M. (2019, July). The Method of Translation of the Diagram with One Type Directed Links into the Inhibitor Petri Net. In *International Conference on Computational Science and Its Applications* (pp. 562-572). Springer, Cham.
- [6] Planas, E., & Cabot, J. (2020). How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus. *Computer Standards & Interfaces*, 67, 103363.
- [7] Anwar, M. W., Azam, F., Khan, M. A., & Butt, W. H. (2019, March). The Applications of Model Driven Architecture (MDA) in Wireless Sensor Networks (WSN): Techniques and Tools. In *Future of Information and Communication Conference* (pp. 14-27). Springer, Cham.
- [8] Nanthaamornphong, A., & Leatongkam, A. (2019). Extended ForUML for Automatic Generation of UML Sequence Diagrams from Object-Oriented Fortran. *Scientific Programming*, 2019.
- [9] Assunção, W. K., Vergilio, S. R., & Lopez-Herrejon, R. E. (2020). Automatic extraction of product line architecture and feature models from UML class diagram variants. *Information and Software Technology*, 117, 106198.
- [10] Zahoor, M., Azam, F., Anwar, M. W., Yousaf, N., & Kashif, M. (2019, July). A UML Profile for the Service Discovery in the Enterprise Cloud Bus (ECB) Framework. In *Conference on Complex, Intelligent, and Software Intensive Systems* (pp. 269-279). Springer, Cham.
- [11] Yildiz, O. (2019). Object-based modeling of restrictions on the sale of agricultural land. *Land use policy*, 82, 538- 549.
- [12] Lozynska, O., Savchuk, V., & Pasichnyk, V. (2019, September). Individual Sign Translator Component of Tourist Information System. In *International Conference on Computer Science and Information Technology* (pp. 593-601). Springer, Cham.
- [13] Ruiz, P., & Dorransoro, B. (2019). A Novel CAD Tool for Electric Educational Diagrams. *Applied Sciences*, 9(4), 810.
- [14] Störrle, H. (2020). Structuring large models with MONO: Notations, templates, and case studies. In *Model Management and Analytics for Large Scale Systems* (pp. 141-166). Academic Press.
- [15] Júnior, A. A., Misra, S., & Soares, M. S. (2019, July). A Systematic Mapping Study on Software Architectures Description Based on ISO/IEC/IEEE 42010: 2011. In *International Conference on Computational Science and Its Applications* (pp. 17-30). Springer, Cham.
- [16] Pathak, N., Sharma, G., & Singh, B. M. (2019). Trusted Operating System-Based Model-Driven Development of Secure Web Applications. In *Software Engineering* (pp. 421-432). Springer, Singapore.
- [17] Abdelhadi, F., Brahim, A. A., & Zurfluh, G. (2019, October). Applying a Model-Driven Approach for UML/OCL Constraints: Application to NoSQL Databases. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 646-660). Springer, Cham.
- [18] Fitzgerald, K., Browne, L. M., & Butler, R. F. (2019). Using the Agile software development lifecycle to develop a standalone application for generating colour magnitude diagrams. *Astronomy and Computing*, 28, 100283.
- [19] Mohanan, M., & Bajwa, I. S. (2019). Requirements to Class Model via SBVR: RECM via SBVR TOOL. *International Journal of Open Source Software and Processes (IJOSSP)*, 10(2), 70-87.
- [20] Chien, C. F., Chen, G. Y. H., & Liao, C. J. (2019). Designing a Connectivist Flipped Classroom Platform Using Unified Modeling Language. *International Journal of Online Pedagogy and Course Design (IJOPCD)*, 9(1), 1-18.

- [21] Bluemke, I., & Malanowska, A. (2019, July). Tool for Assessment of Testing Effort. In *International Conference on Dependability and Complex Systems* (pp. 69-79). Springer, Cham.
- [22] Traoré, M. K. (2019). *Unified Approaches to Modeling. In Model Engineering for Simulation* (pp. 43-56). Academic Press.
- [23] Wang, Y., & Bai, L. (2019). Fuzzy Spatiotemporal Data Modeling Based on UML. *IEEE Access*, 7, 45405-45416.
- [24] Mumtaz, H., Alshayeb, M., Mahmood, S., & Niazi, M. (2019). A survey on UML model smells detection techniques for software refactoring. *Journal of Software: Evolution and Process*, 31(3), e2154.
- [25] Korableva, O. N., Kalimullina, O. V., & Mityakova, V. N. (2019, July). Designing a System for Integration of Macroeconomic and Statistical Data Based on Ontology. In *Intelligent Computing-Proceedings of the Computing Conference* (pp. 157-165). Springer, Cham.
- [26] Jovic, S., Cukaric, A., Raicevic, A., & Tomov, P. (2019). Assessment of electronic system for e-patent application and economic growth prediction. *Physica A: Statistical Mechanics and its Applications*, 520, 93-105.
- [27] Bocciarelli, P., D'Ambrogio, A., Falcone, A., Garro, A., & Giglio, A. (2019). A model-driven approach to enable the simulation of complex systems on distributed architectures. *simulation*, 0037549719829828.
- [28] Tekinerdogan, B., Babur, Ö., Cleophas, L., van den Brand, M., & Akşit, M. (2020). Introduction to model management and analytics. In *Model Management and Analytics for Large Scale Systems* (pp. 3-11). Academic Press.
- [29] Baklouti, A., Nguyen, N., Mhenni, F., Choley, J. Y., & Mlika, A. (2019). Improved Safety Analysis Integration in a Systems Engineering Approach. *Applied Sciences*, 9(6), 1246.
- [30] Mehryar, S., Sliuzas, R., Schwarz, N., Sharifi, A., & van Maarseveen, M. (2019). From individual Fuzzy Cognitive Maps to Agent Based Models: Modeling multi-factorial and multi-stakeholder decision-making for water scarcity. *Journal of environmental management*, 250, 109482.