|  |
| --- |
|  |
| **SDLC:**  SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.  The following figure is a graphical representation of the various stages of a typical SDLC.  A typical Software Development Life Cycle consists of the following stages −  Stage 1: Planning and Requirement Analysis  Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.  Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.  Stage 2: Defining Requirements  Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.  Stage 3: Designing the Product Architecture  SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.  This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.  A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.  Stage 4: Building or Developing the Product  In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.  Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.  Stage 5: Testing the Product  This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.  Stage 6: Deployment in the Market and Maintenance  Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).  Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.  SDLC Models  There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.  Following are the most important and popular SDLC models followed in the industry −   * Waterfall Model * Iterative Model * Spiral Model * V-Model * Big Bang Model |
| **Observation & Learning:**  **HEALTH PREDICTION SYSTEM**  Need powerful set of tools that has the potential to assist in the planning and coordination of the health care services. The project was designed to reflect the System Development Life Cycle (SDLC) development methodology, which consists of five stages: planning, analysis, detailed system design, implementation, and support. The SDLC describes activities and functions that all systems developers perform, regardless of which approach they use .  Many have proposed the use of electronic personal health record systems (PHRs) in the self-management process, but adoption remains low. The purpose of this research was to validate and test an explanatory model of the barriers and facilitators to people adoption of personal health records for self-managing chronic illnesses. The long range goal of the research is to use the explanatory model to develop interventions that will maximize the facilitators and minimize the barriers to adoption.  **1.REQUIREMENT:**   * **Patient Login: -** Patient Login to the system using his ID and Password. * **Patient Registration: -**If Patient is a new user he will enter his personal details and he will user Id and password through which he can login to the system. * **Admin Login:** Admin can login to the system using his ID and Password. * **Doctor Login: -** Doctor will access the system using his User ID and Password.   .  **2.PLANNING**  **Design:** The user will get 2 login page i.e. patient and doctor . The user will select its best suitable option. As the user will login with his/her given id password a user oriented page will be open and they can view their pervious tasks and personal details in that.  **Functionality:** It might have happened so many times that you or someone yours need doctors help immediately, but they are not available due to some reason. The Health Prediction system is an end user support and online consultation project. Here we propose a system that allows users to get instant guidance on their health issues through an intelligent health care system online. The system is fed with various symptoms and the disease/illness associated with those systems. The system allows user to share their symptoms and issues. It then processes user’s symptoms to check for various illnesses that could be associated with it. Here we use some intelligent data mining techniques to guess the most accurate illness that could be associated with patient’s symptoms. In doctor module when doctor login to the system doctor can view his patient details and the report of that patient. Doctor can view details about the patient search what patient searched for according to their prediction. Doctor can view his personal details. Admin can add new disease details by specifying the type and symptoms of the disease into the database. Based on the name of the disease and symptom the data mining algorithm works. Admin can view various disease and symptoms stored in database. This system will provide proper guidance when the user specifies the symptoms of his illness.  **I/P AND O/P:** The input to login the page. For patient he can search for his/her health issues according to the symptoms he/she is facing and he can type the name of doctor and directly contact to that doctor. For doctorhe/she can view the sympotoms of patient which patient has typed abd can easily give suggestion and medicine to that patient.  Both doctor and patient can contact each other.  **3.MODELING:**   * **My Details: -** Patient can view his personal details. * **Disease Prediction: -** Patient will specify the symptoms caused due to his illness. System will ask certain question regarding his illness and system predict the disease based on the symptoms specified by the patient and system will also suggest doctors based on the disease. * **Search Doctor:-** Patient can search for doctor by specifying name, address or type. * **Patient Details:** Doctor can view patient’s personal details. * **Notification:** Doctor will get notification how many people had accessed the system and what all are the diseases predicted by the system. * **View Doctor:** Admin can view various Doctors along with their personal details. * **View Disease:** Admin can view various diseases details stored in database. * **View Patient:** Admin can view various patient details who had accessed the system. * **View Feedback:** Admin can view feedback provided by various users * **Add Doctor:** Admin can add new doctor details into the database. * **Add Disease:** Admin can add disease details along with symptoms and type. * **Feedback:-**Patient will give feedback this will be reported to the admin.   **4.Coding and Testing:**  Can be written in any programming language and required platform to host the given application.  **Software Requirements:**   * Windows 7 and above * Microsoft SQL Server * Visual Studio   **Hardware Components:**   * Processor – Dual Core * Hard Disk – 50 GB * Memory – 1GB RAM   **Various Testing methods can be used like:**  **Unit testing**  **Integration testing**  **Validation testing**  **5. Devlopement:**  Required engineering for developing project on suitable platform and verification and validation is required according to the needs mentioned for project.  **Advantages:**   * User can search for doctor’s help at any point of time. * User can talk about their illness and get instant diagnosis. * Doctors get more clients online.   **Disadvantages:**   * The system is not fully automated; it needs doctors for full diagnosis. |
| **Questions:**   1. **Explain the shortcomings of the classical waterfall model.**  * **No feedback path:** In classical waterfall model evolution of a software from one phase to another phase is like a waterfall. It assumes that no error is ever committed by developers during any phases. Therefore, it does not incorporate any mechanism for error correction. * **Difficult to accommodate change requests:** This model assumes that all the customer requirements can be completely and correctly defined at the beginning of the project, but actually customers’ requirements keep on changing with time. It is difficult to accommodate any change requests after the requirements specification phase is complete. * **No overlapping of phases:** This model recommends that new phase can start only after the completion of the previous phase. But in real projects, this can’t be maintained. To increase the efficiency and reduce the cost, phases may overlap.  1. Describe phase entry and phase exit criteria of each phase in classical waterfall model.   Classical waterfall model divides the life cycle into a set of phases. This model considers that one phase can be started after completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus the development process can be considered as a sequential flow in the waterfall. Here the phases do not overlap with each other. The different sequential phases of the classical waterfall model are shown in the below figure: https://media.geeksforgeeks.org/wp-content/uploads/Screenshot-1-3.png |
| **Conclusions:**  Software Development Life Cycle (SDLC) and analyse Health Prediction System conducted as a part of various phases. |