|  |
| --- |
| **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   **PROJECT (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  **INTENDED AUDIENCE AND READING SUGGESTIONS**  Smart Health Prediction System allows users to get instant guidance on their health issues through an health care system online. The system is fed with various symptoms and the diseases associated with that system. The system is first taught with various symptoms and the disease associated with each system. The user can also contact to doctor and take advice and prescription. The Health Prediction system is an end user support and online consultation project. According to this project their is no age limit anyone male or female children or old age people can use this app. This app will be designed very user friendly. Any one who wants to resolve their small health related issues which are normally ignored by people can be easily resolved.  **PRODUCT SCOPE**  A huge part of this involves scanning and storing documents on a per patient basis.The training data should be updated according to the research and any other new disease found in environment . Each document may have different pieces of data that need to be tracked in the database.. Session notes, and individual session notes data should be kept and should be removed after the problem is resolved. More specifics on data to be stored can be found in the history database. The portal has some major parts: patients and doctor,sessions.  Once the people will understand the importance of health care suggestion we can add billing system.  **Permission:**  Location, camera, microphone,phone application, contacts,access media are required for the application to work properly.  **PRODUCT FUNCTION**  This application is able to predict all the types of disease and predict the health of the user. The user can contact to any doctor sitting at home. Smart health prediction will predict the user if their is any health issue and any precaution they have to take after they pass the test.  **FR**  A proper login system will be provided to each user of the system so as to access their account and history.  Administrator will be able to add/delete/edit any system user and make changes according to the feedback send by user.  The doctor can access to previous history of the patient and can call or message according to their need.  Message should be send to the patient when doctor is free  **NFR**   1. Smart Health Prediciton system will be having a user friendly interface. 2. Smart Health Prediciton system will provide all the necessary data regarding the disease , treatment, doctor and other instruction. 3. Smart Health Prediciton system will have all the doctors team related to all the disease and treatment. 4. A staff database that will be available that will collect samples from house. 5. Smart Health Prediciton system will be providing you high security with assurance of no leakage of your personal data and credentials. 6. Smart Health Prediciton system will be budget friendly. 7. Smart Health Prediciton system will be scalable and flexible and will be open to updates from time to time with more security and efficient functions. 8. Administrator module will allow the administrator to administer the different aspects of the system such as maintaining the record of users.   **HARDWARE INTERFACES**  OS: Will work with Windows, MacOS and Andriod   1. If user is using the windows version of the software then user must be having a laptop 2. If user is using the Android version of the software then user must be having a mobile phone.  * Processor – Dual Core * Hard Disk – 50 GB * Memory – 1GB RAM   **Various Testing methods can be used like:**  **Unit testing**  **Integration testing**  **Validation testing**  **SOFTWARE INTERFACES**  **Software Requirements:**   * Windows 7 and above * Microsoft SQL Server * Visual Studio   **WORKING**  **C:\Users\abc\Desktop\sem6\se\1.jpg**  **INTERFACE**  **C:\Users\abc\Desktop\sem6\se\2.jpg**  C:\Users\abc\Desktop\sem6\se\3.jpg  C:\Users\abc\Desktop\sem6\se\4.jpg |
| **Observation & Learning:**  We learned and observed the system requirement sepecifications for the project. |
| 1. **Give the difference between Functional & Non-functional requirements with suitable example.**  | **Functional Requirements** | **Non Functional Requirements** | | --- | --- | | 1. A functional requirement defines a system or its component. | A non-functional requirement defines the quality attribute of a software system. | | 1. It specifies “What should the software system do?” | It places constraints on “How should the software system fulfill the functional requirements?” | | 1. Functional requirement is specified by User. | Non-functional requirement is specified by technical peoples e.g. Architect, Technical leaders and software developers. | | 1. It is mandatory. | It is not mandatory. | | 1. It is captured in use case. | It is captured as a quality attribute. | | 1. Defined at a component level. | Applied to a system as a whole. | | 1. Helps you verify the functionality of the software. | Helps you to verify the performance of the software. | | 1. Functional Testing like System, Integration, End to End, API testing, etc are done. | Non-Functional Testing like Performance, Stress, Usability, Security testing, etc are done. | | 1. Usually easy to define. | Usually more difficult to define. | | **Example**  **1)** Authentication of user whenever he/she logs into the system. **2)** System shutdown in case of a cyber attack. **3)** A Verification email is sent to user whenever he/she registers for the first time on some software system. | **Example**  **1)** Emails should be sent with a latency of no greater than 12 hours from such an activity. **2)** The processing of each request should be done within 10 seconds **3)** The site should load in 3 seconds when the number of simultaneous users are > 10000 |  1. **Identify FR and NFR from the below problem statement:**   **Internet has led to discussion of e-democracy and online voting. Many people’s think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make’s use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting**  **E-Voting Machine**  **Functional Requirements:**   1. **Providing Authentication means**: Here the user is provided with the authentication means required for the elector to cast a vote such as a voting id. 2. **Authentication of the elector**: Here the user will be asked to upload a picture of his voting id or any other authentication means given for the verification of the elector. 3. **Managing Parties**: Here the list of the parties will be provided to the user with all the necessary information about the parties for the ease of the user to cast his vote. 4. **Preview for Vote**: Here the user will be provided with a preview of casting a vote to avoid confusion. 5. **Casting votes**: Here the user will be asked to cast the vote under high security and protection environment. 6. **Counting Votes**: Here augmentation of the election result will be done. This system use case deals with the final tally calculation. 7. **Verifying Result Integrity**: This use case serves the request for the verification of the procedure integrity and to verify the election result.   **Non Functional Requirements:**   1. **Performance Requirements:** The user must have all the necessary performance requirements required by the software. 2. **Safety Requirements:** Here all the personal credentials of the user will be protected and secured. 3. **Security Requirements:** Here the vote casted by the user will be secured and the identity will not be revealed. 4. **User Documentation:** Here the software privacy agreement and the user documentation will be given. 5. **User support and Guide:** In case of any technical issue or any other issue proper customer support and guide will be given to the user. |