MEDICAPS UNIVERSITY, INDORE

Department of Computer Science & Engineering
FACULTY OF ENGINEERING

Practical File

Software Engineering [CS3CO26]

Submitted To:
Ms Priyanka Dhasal

Submitted By:
Dhananjay Porwal
EN19CS301110

Index

Practical-1

Study of various software with their examples.

S.No	Name	Domain	Year	Key Feature
1	Google Assistant	Utility Software	2021	a) Real-time spoken translations. b) Open apps on your phone. c) Make appointments and send messages.
2	Visual Studio Code	Education Software	2021	a) Command Line. b) Git Integration. c) Split view.
3	Microsoft Edge	Utility Software	2021	a) Install Websites as Apps. b) Web Capture. c) Put Tabs to Sleep.
4	Microsoft Teams	Communication Software	2021	a) Conversations within channels and teams. b) Document storage in SharePoint. c) Online video calling and screen sharing.
5	Notes Speech	Education Software	2020	a) Fast and accurate Speech Recognition. b) Works on any page, including gmail and others. c) Keyboard shortcut to start &pause recognition.
6	Notion	Education Software	2021	a) Powerful Note Taking. b) Tasks and Projects. c) Integration.
7	Telegram	Communication Software	2021	a) Auto delete messages. b) New animated emojis. c) Broadcast groups.
8	Ubuntu 20.04 LTS	System Software	2021	a) Easy and simple to use. b) Long Term Support releases.
9	VLC Media Player	Multimedia Software	2020	a) Use VLC as a Video Downloader for YouTube. b) Convert Videos to Any Format. c) Play Internet Radio and Podcasts in VLC.
10	Windows 11	System Software	2021	a) Integrated Android apps. b) Xbox tech for better gaming.

Practical-2

Problem Statement for any case study project.

GoWin

Health is the greatest wealth for every living creature. In this pandemic situation, almost all the work shifted on the technology. Not everyone is familiar with the latest technologies. As a result, People pay for searching COVID related resources. **Thus, we make a user-friendly, easy-to-use, responsive and open-source project with all the COVID-19 related resource at one place which you can access anywhere anytime. GoWin, A platform independent Web-Application which brings all the medical resources at one place with clear and minimal UI/UX.** It provides information about COVID bed availability, ICU, Plasma, Ambulance, nearest vaccination center, free-food, Realtime-vaccination, smart-phone-app, safety measures, common assumptions, helpline number and many more...

This is responsive web-app, which is extreme lite weight and doesn't save any cookies on your machine. NO need of any special permission to access this web-app. This is made in the way by which we can easily update and upgrade it without any hesitation. Currently we use HTML5, CSS3, JavaScript, ReactJs, JavaQuery, MySQL, Python, and several Government-own-API which are time to time updated with the help of our Machine Learning Bot. We use some open-source illustration to make it more user friendly and interactive.

Our future goals are:

1. Making separate section for GREEN CORRIDOR REQUEST.
2. Consumer-Forum section.
3. Government based Schemes for Pandemic.

Practical-3

Functional and non-functional requirements

In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

We'll focus on the following set of requirements while designing GoWin Community App.

Functional Requirements

1. The system bot will update API time-to-time.
2. User should get the best available information according to their requirement.
3. Quick Helpline support.
4. Latest News related to COVID-19.
5. Free Meals near your for COVID-Warriors.
6. Vaccination-Center near you.
7. Dark Theme/Light Theme.
8. All the medical information/facilities updated in every 7 minutes.

Non-Functional Requirements

1. Low Vulnerability.
2. Independent of screen size.
3. No Cookies generated.
4. Easily upgradable with new technologies.
5. Version-Control-System.
6. Open-Source.
7. Access by any location, even outside of INDIA.
8. No special permission required.

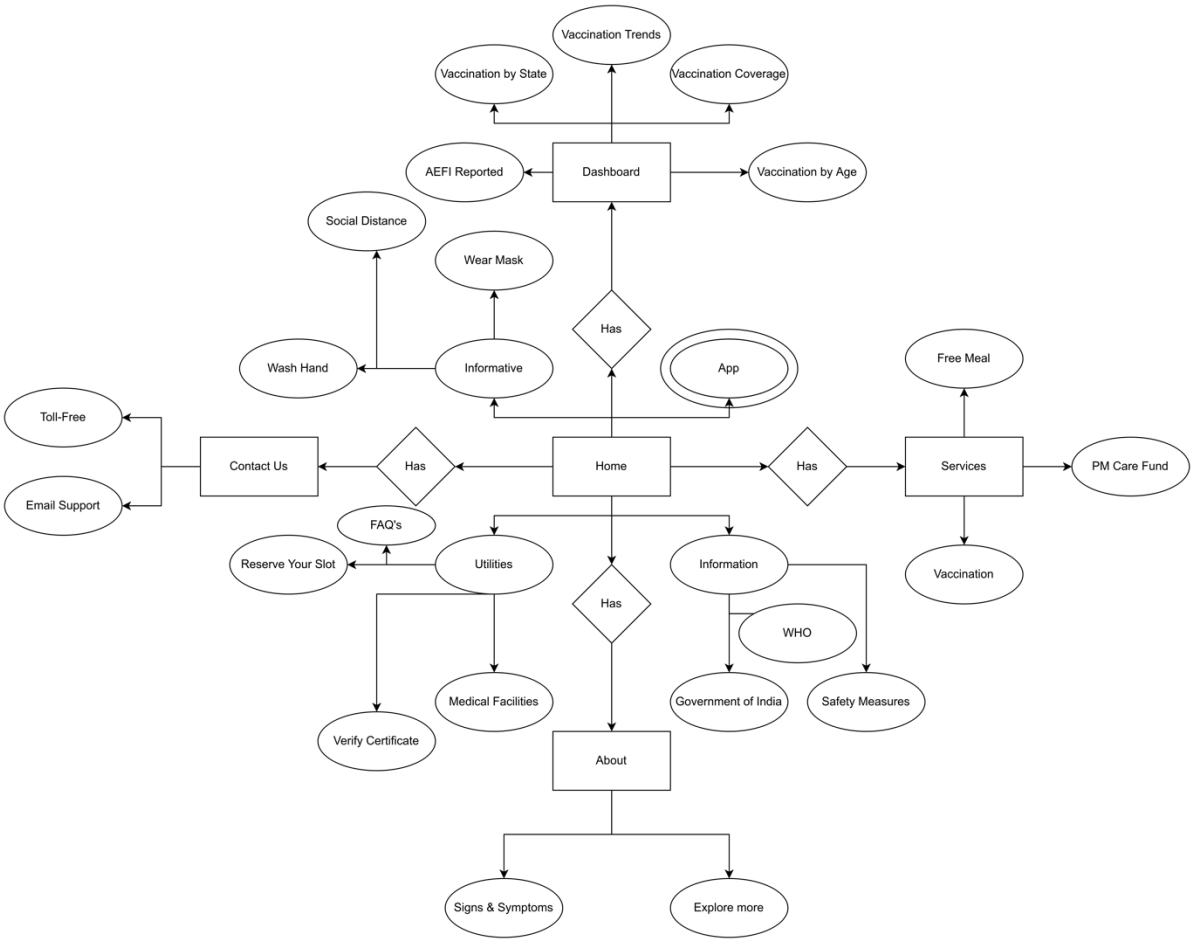
Practical-4

ER Diagram of Case Study Project

ER-modelling is a data modelling method used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modelling method are called Entity-Relationship Diagrams or ER diagrams or ERDs.

1. Entity
An entity can be a real-world object, either animate or inanimate, that can be merely identifiable. An entity is denoted as a rectangle in an ER diagram.
Entities – Home, About, Services, Contact-Us, Dashboard, App.

2. Attributes
Entities are denoted utilizing their properties, known as attributes. All attributes have values.
Attributes example in this diagram – Social Distance, AEFI Report, Informative, Toll-Free-Number, Email Support.



Practical-5

Data Flow Diagram (DFD) of Case Study Project

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information 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. Specific operations based on the data can be represented by a flowchart.

Levels in Data Flow Diagrams (DFD)

The DFD may be used to perform a system or software at any level of abstraction. Infact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily three levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

0-level DFD

It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows.

1-level DFD

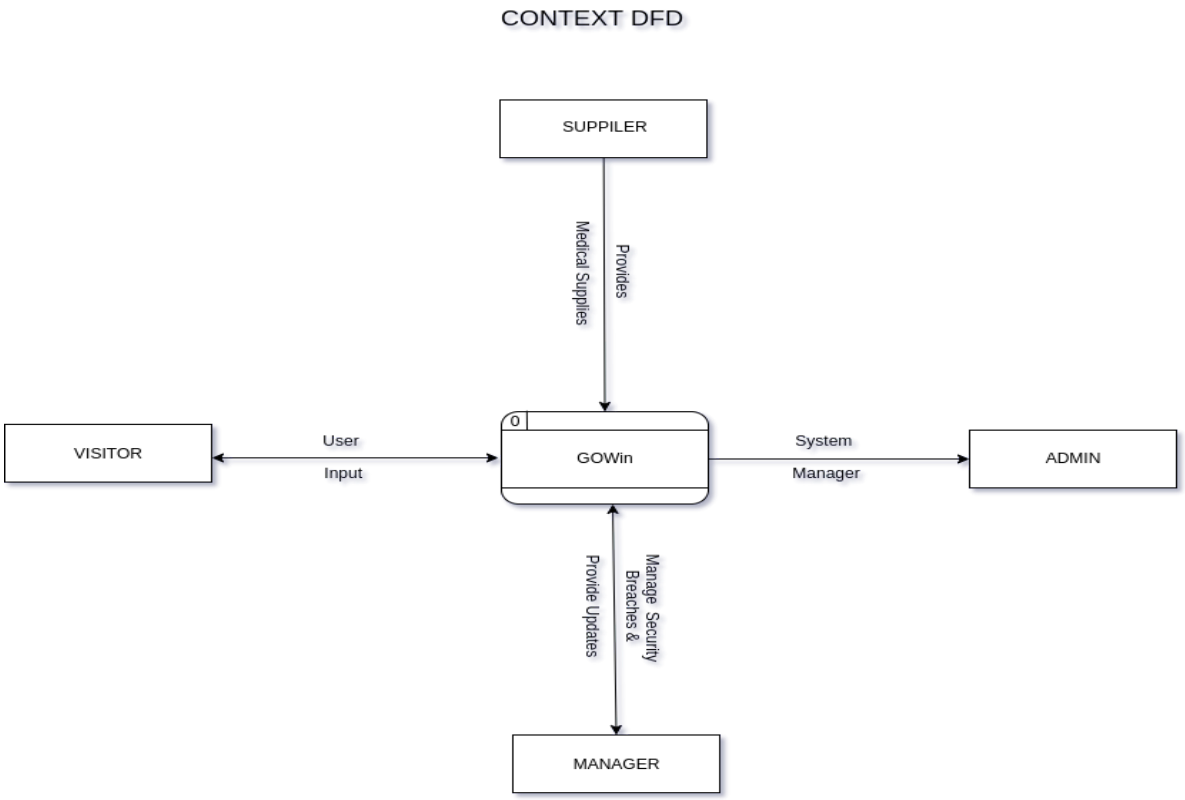
In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses.

2-Level DFD

2-level DFD goes one process deeper into parts of 1-level DFD. It can be used to project or record the specific/necessary detail about the system's functioning.

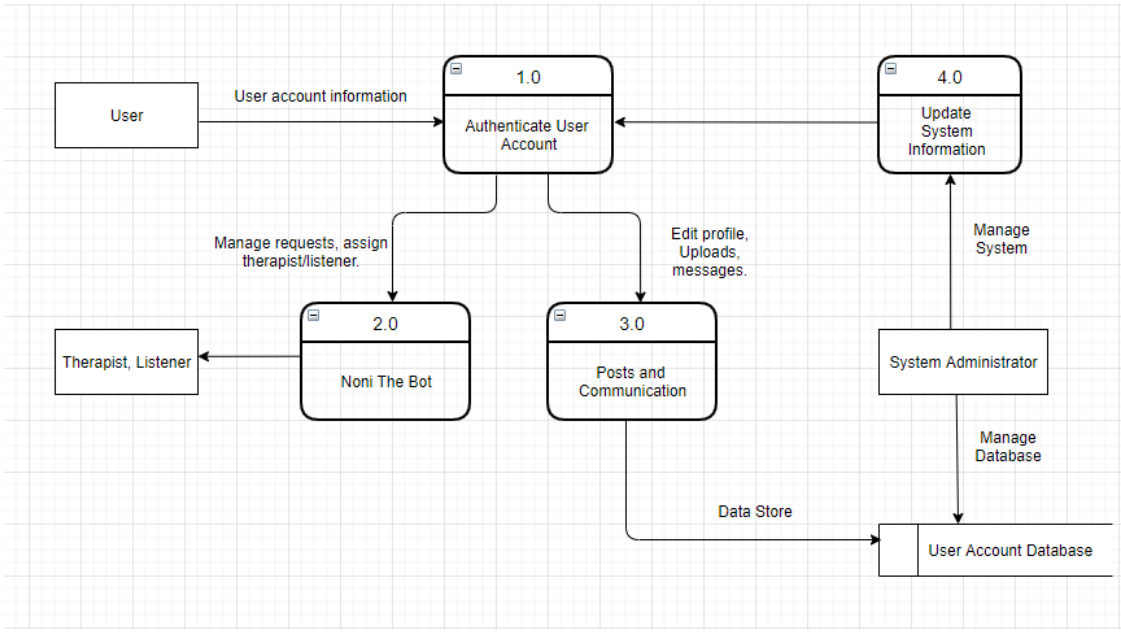
Zero level data flow diagram(0 level DFD) of GoWin:

The zero level data flow diagram shows three sources of data within the system: the User(s), Social Networking Application- GoWin, and System Admin. Users provide information as input, the system generates all information and services as output to the users.



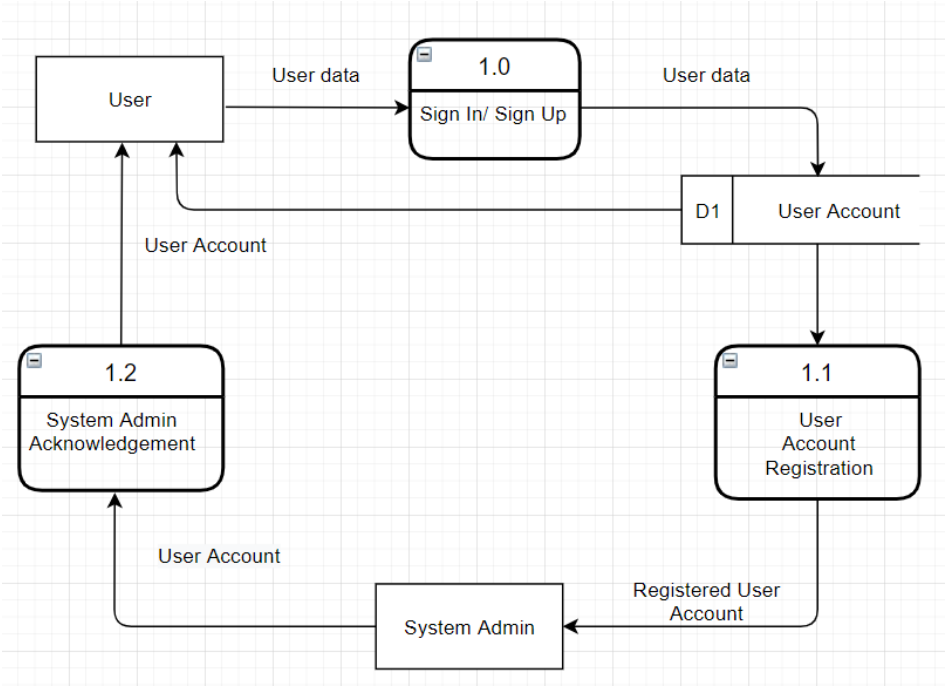
First level data flow diagram (1 level DFD) of GoWin:

This diagram represents the GoWin Platform major processes, data flow and data stores. There are three streams or flows of data. The user information input for registration is being stored in user account database.

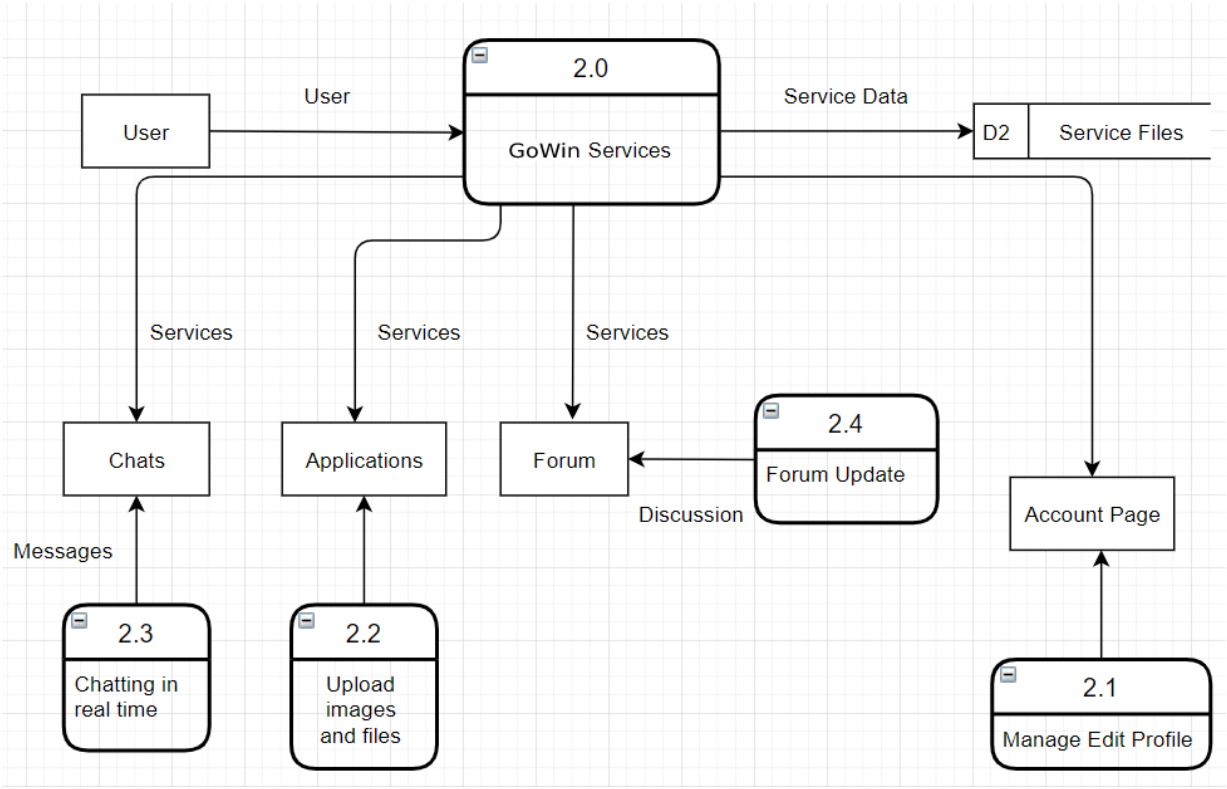


Second level Data flow diagram (2 level DFD) of GoWin:

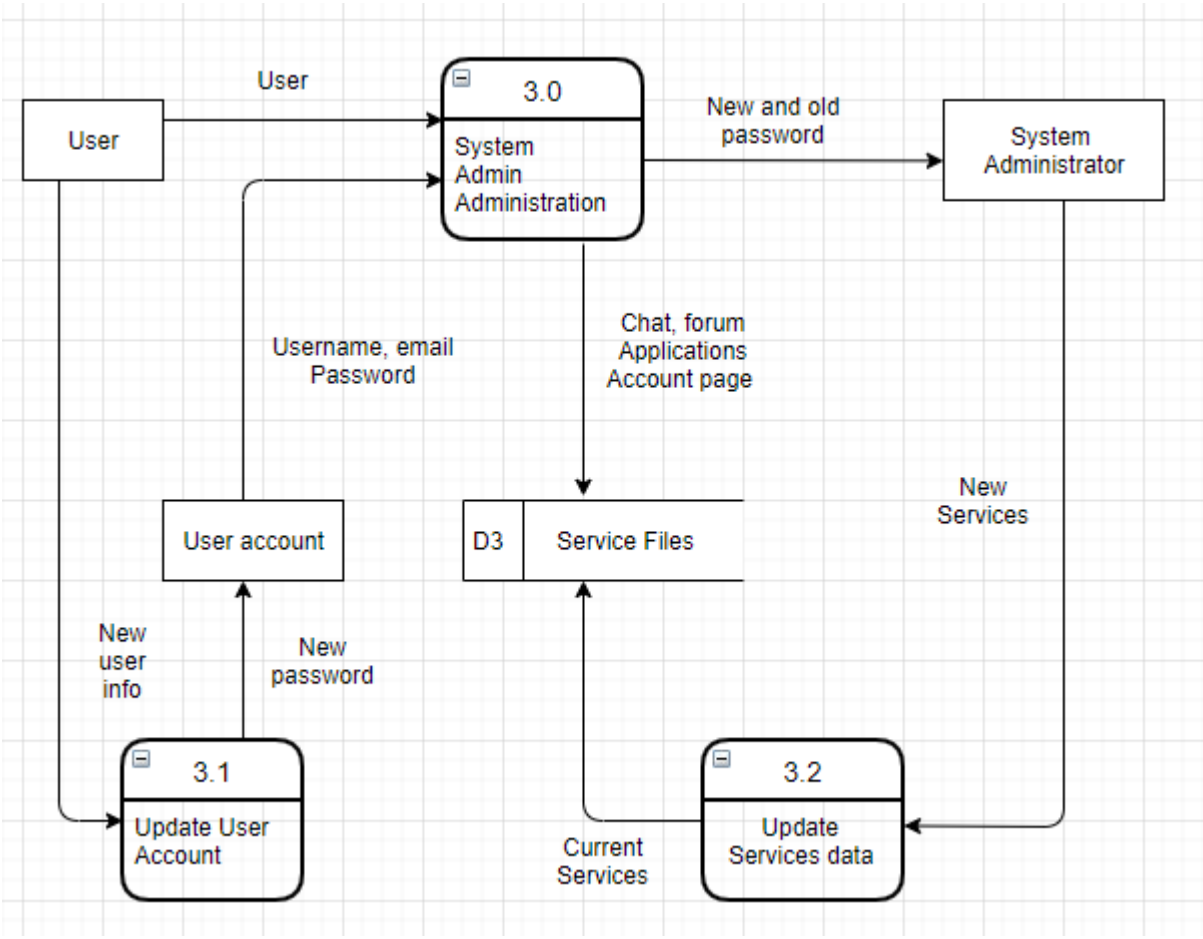
This diagram shows the decomposition process 1.0 online registration from level 0 diagram of Social Networking Application. The user will register online. Once the user is registered, the System Admin has the authority to verify and update all user account being entered.



This diagram shows the decomposition process 2.0 the Online Services. The user can use the services of Social Networking Application like chat: sending messages in real time upload images and files. The user can also manage his/ her profile.



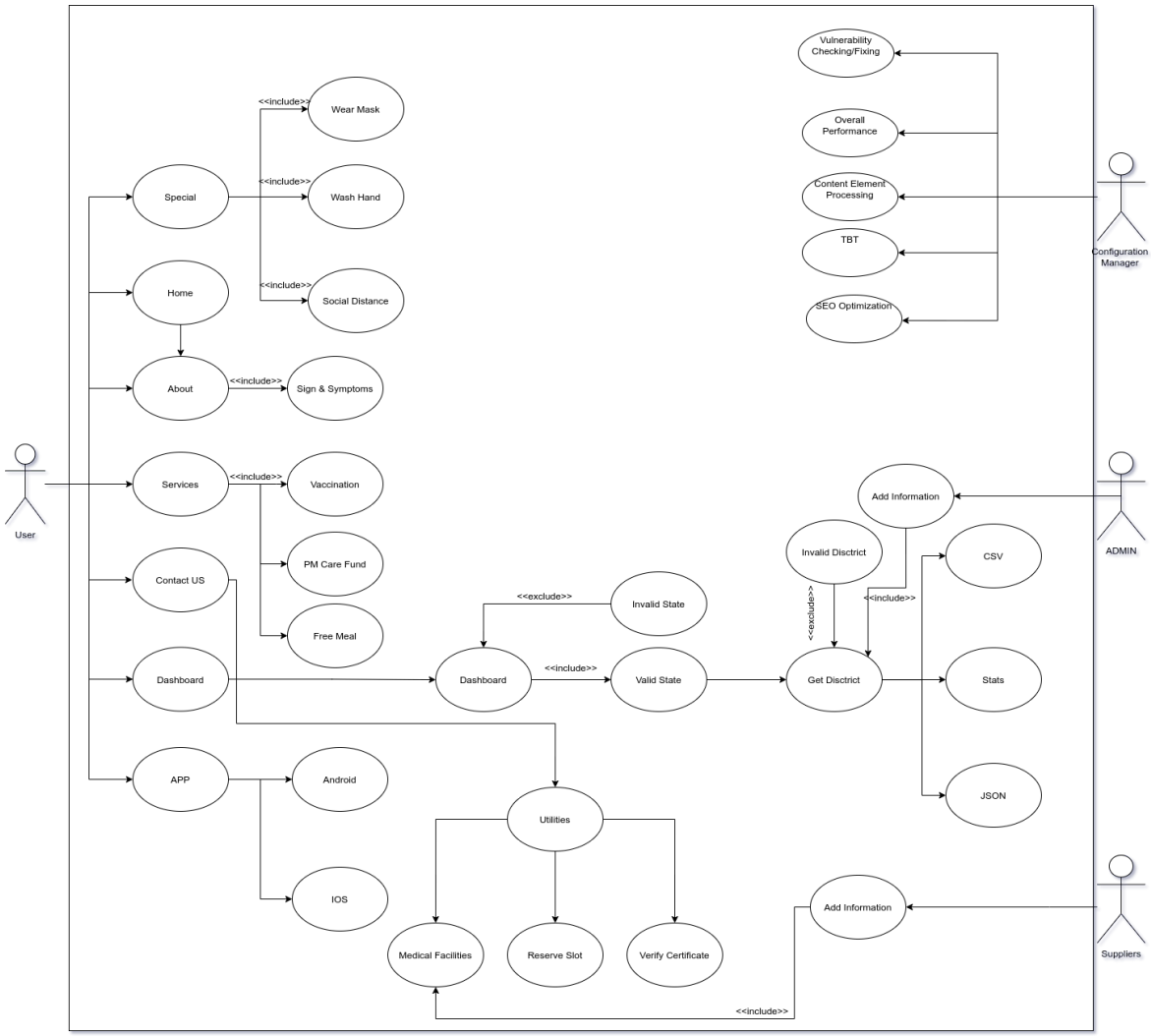
This diagram shows the decomposition process 3.0 which is System Admin Maintenance. The system administrator is responsible to update the user account and services then, acknowledge all the changes of the user to his/ her own profile or account.



Practical-6

Use case diagram of Case Study Project

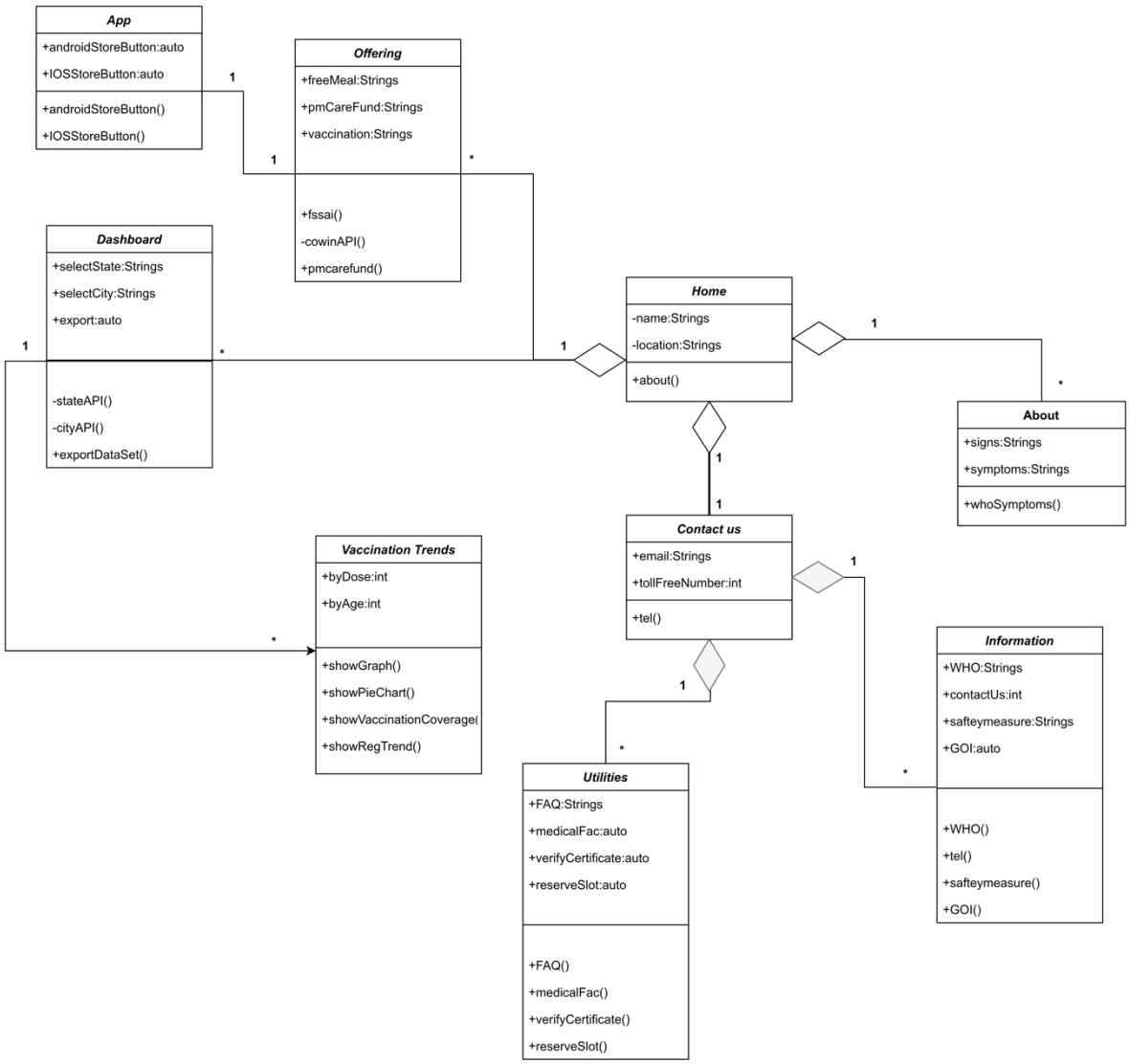
Use Case Diagram:- A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well.



Practical-7

Class diagram of Case Study Project

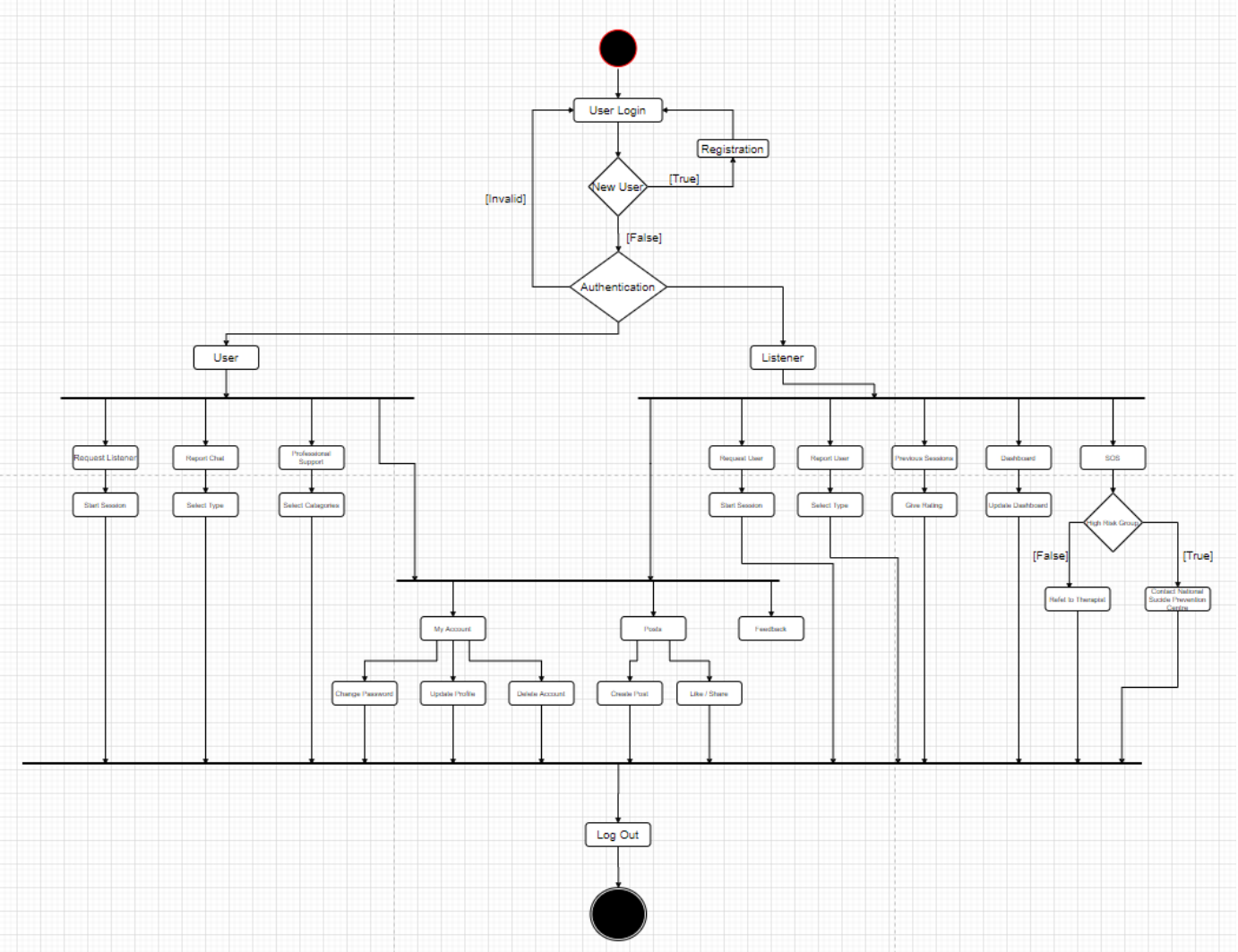
A class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. In our diagram the class user and administrator is the combined login which leads us to the account. The account class can be used to view profile and jobs like sending friend request and reporting profile. Other than this, registrations are done on the account. Using profile class, we can send and delete messages and also requests the scheduler class to pass the message and other information including issues of the user. After the meeting being scheduled by the scheduler class, the therapist class has all the information related to the patient and they provide consultations, keep records of the patients and if the situation is out of hand, suicide prevention helpline is contacted at the earliest. Unlike the user whose identity is anonymous, the therapist is known and has a proper therapist id, name and email.



Practical-8

Activity diagram of Case Study Project

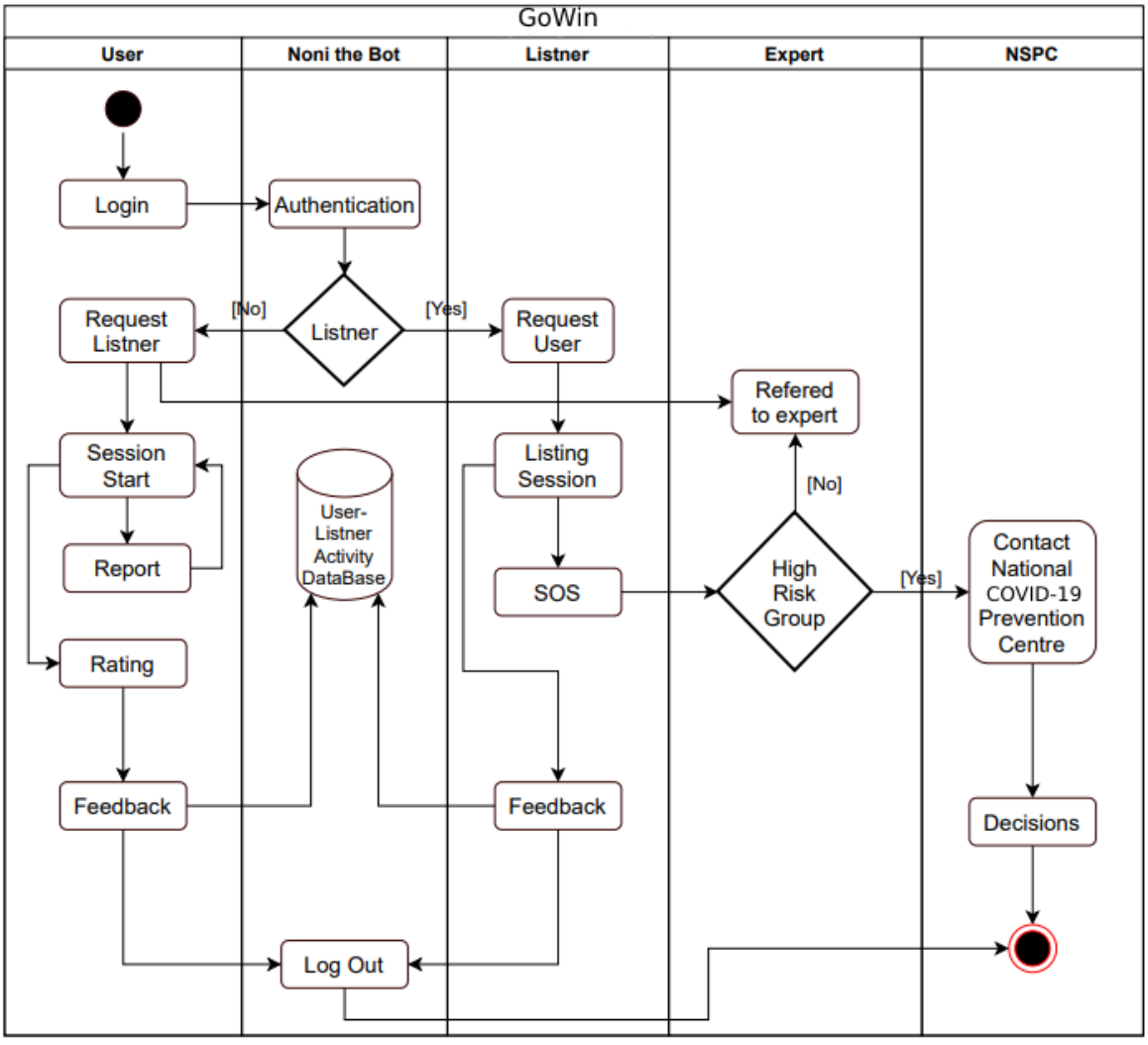
Activity Diagram:- Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram is essentially a flowchart that shows activities performed by a system.



Practical-9

Swimlane diagram of Case Study Project

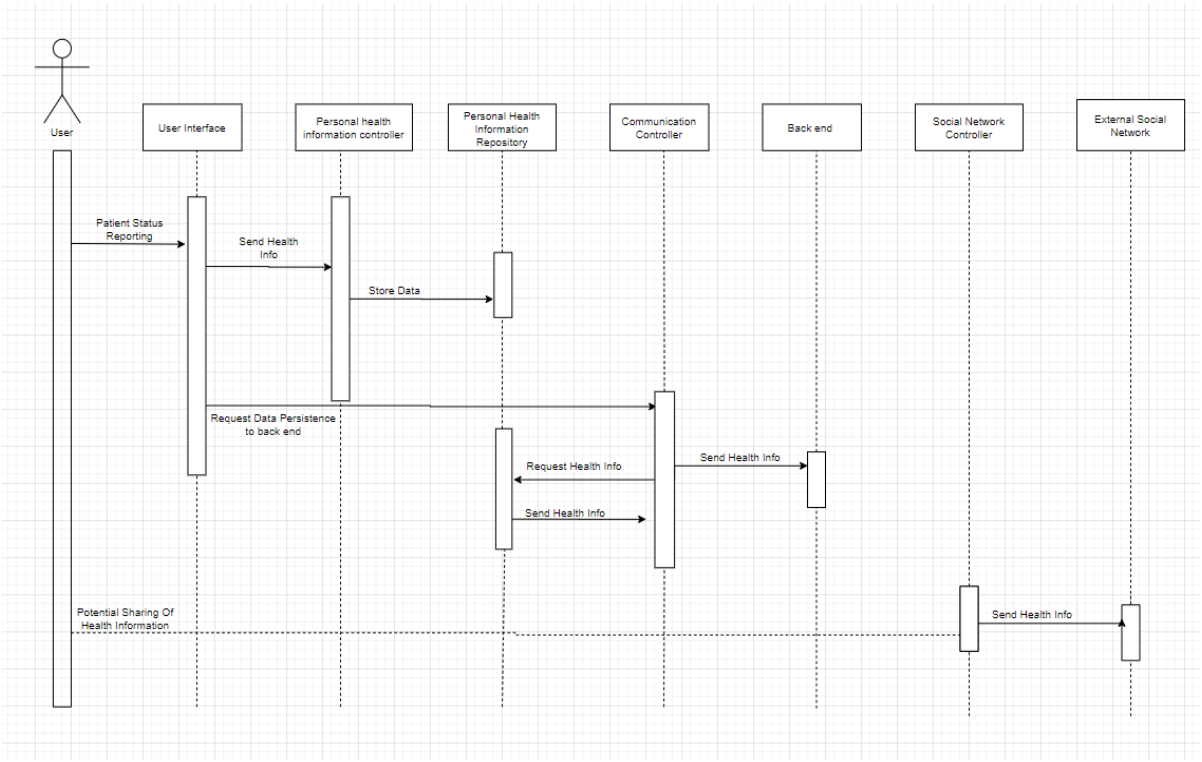
Swim Lane Diagram:- Swim lane diagrams are flowcharts that show both a process from start to finish and who is responsible for each step in the process.



Practical-10

Sequence diagram of Case Study Project

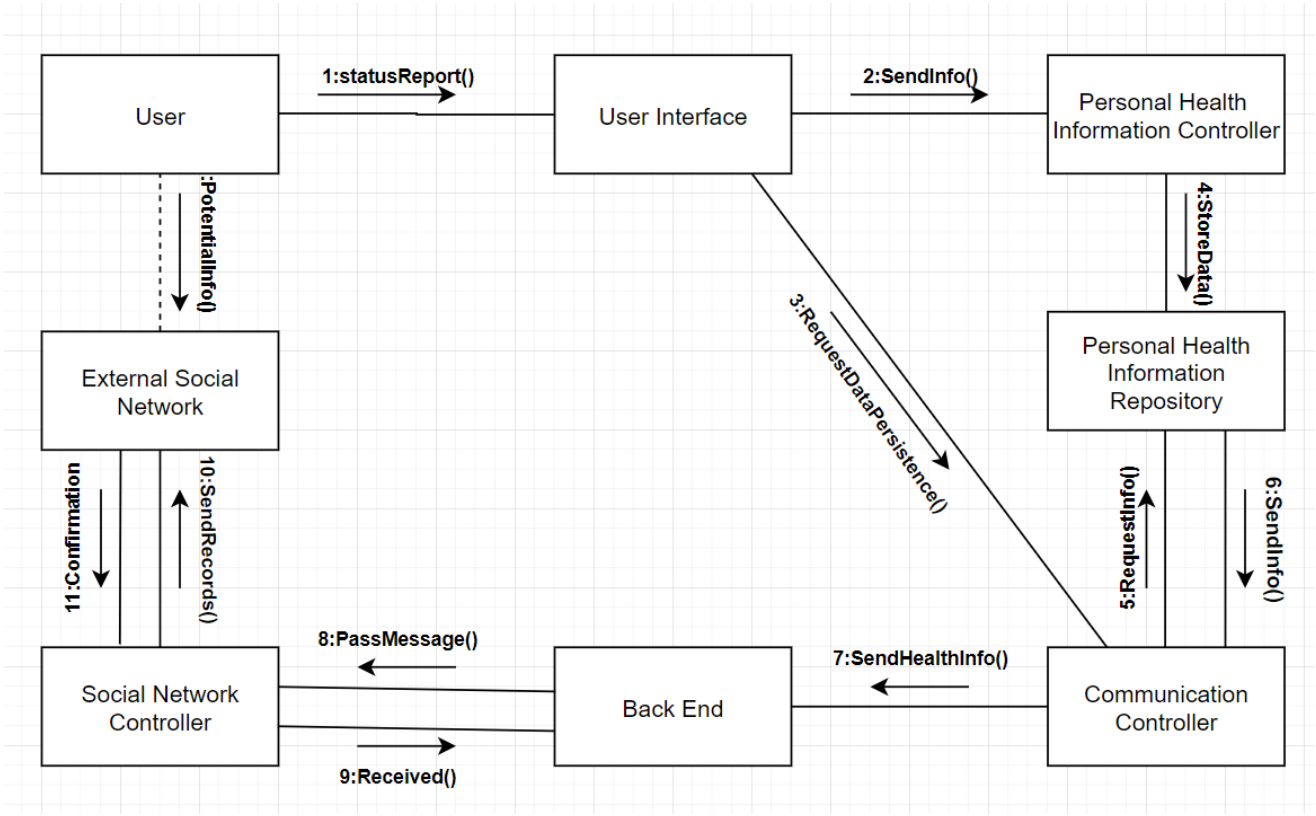
A sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. In our diagram, user has to first go through the user interface for the status reporting. The user interface sends health information to the personal health information controller and requests data persistence to back end and the data is stored in personal health information repository. Then the communication controller requests the repository for the health information and the information is sent to the controller. The health information is eventually sent to back end and the social network controller finally pass the information to the internal social network.



Practical-11

Collaboration diagram of Case Study Project

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Our diagram represents sharing the health info and final records through communication controller to social network controller and eventually the external network. This is the communication medium starting from patient's report to final records to the respective network.



Practical-12

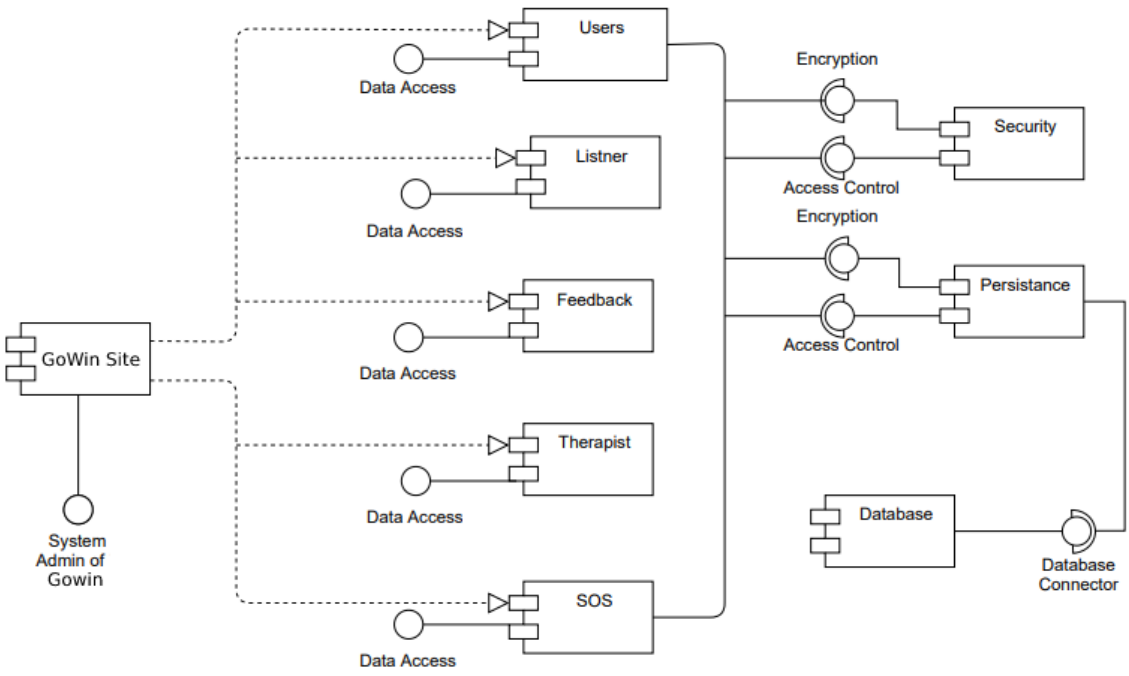
Component diagram for the system

Component diagrams are used to visualize the organization of system components and the dependency relationships between them. They provide a high-level view of the components within a system.

The components can be a software component such as a database or user interface; or a hardware component such as a circuit, microchip or device; or a business unit such as supplier, payroll or shipping.

Component diagrams

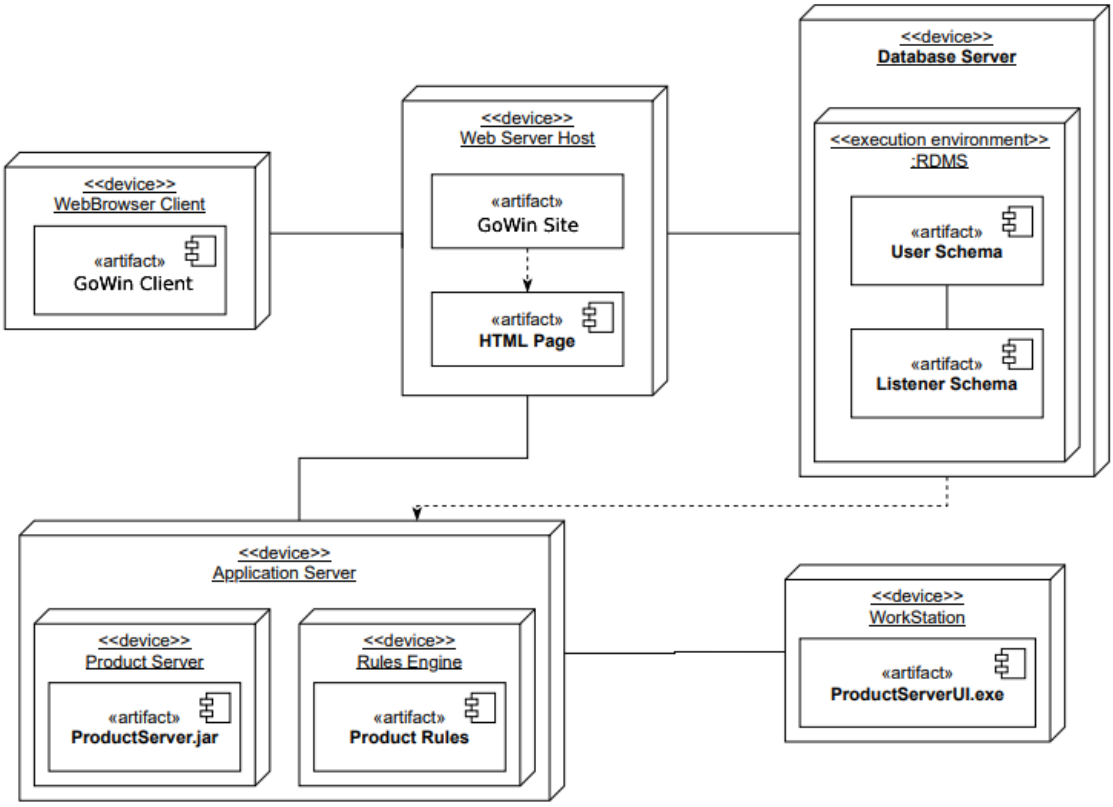
- Are used in Component-Based-Development to describe systems with Service-Oriented-Architecture
- Show the structure of the code itself
- Can be used to focus on the relationship between components while hiding specification detail
- Help communicate and explain the functions of the system being built to stakeholders



Practical-13

Deployment diagram for the system

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.



Practical-14

Study of various testing tools

Software testing tools are required for the betterment of the application or software.

That's why we have so many tools available in the market where some are open-source and paid tools.

The significant difference between open-source and the paid tool is that the open-source tools have limited features, whereas paid tool or commercial tools have no limitation for the features. The selection of tools depends on the user's requirements, whether it is paid or free.

1) TestRail

TestRail is your source for scalable, customizable, web-based test case management. Set up in just minutes with our cloud-based/SaaS solution, or install on TestRail your own server.

- Efficiently manage manual and automated test cases, plans, and runs.
- Get real-time insights into testing progress with informative dashboards, metrics, and activity reports.
- Boost efficiency with milestones, personal to-do lists, and email notifications.
- Document test cases with screenshots and expected results. Use the flexible built-in templates or create your own custom templates.
- Integrate with tools in your CI/CD/DevOps pipeline including JIRA, Bugzilla, Jenkins, TFS and more.
- Enterprise edition designed for large teams & mission-critical projects.
- Support for Docker containers.

2) Testpad

Testpad is a simpler and more accessible manual test tool that prioritises pragmatism over process. Instead of managing cases one at a time, it uses checklist-inspired test plans that can be adapted to a wide range of styles including Exploratory testing, the manual side of Agile, syntax highlighted BDD, and even traditional test case management.

Key features:

- Guest testers, invited by email, who don't need accounts
- Simple enough to use by non-testers; get everyone to help at release time
- Keyboard-driven editor with a javascript-powered (i.e. responsive) UI
- Drag'n'drop organisation of test plans
- Add new tests during testing, as you think of new ideas
- Lightweight integration with issue trackers, including JIRA

3) Xray

Xray is the #1 Manual & Automated Test Management App for QA. It's a full-featured tool that lives inside and seamlessly integrates with Jira. Its aim is to help companies improve the quality of their products through effective and efficient testing.

Features:

- Traceability between requirements, tests, defects, executions
- Define reusable preconditions and associate to tests
- Organize tests in folders and test sets
- Test plans for tracking progress
- Test environments
- BDD – Write Cucumber scenarios in Jira
- Integrates with test automation frameworks (Selenium, JUnit, Nunit, Robot, …)
- Built-in REST API
- CI integrations (Bamboo, Jenkins)
- Built-in reports

4) Practitest

PractiTest is an end-to-end test management tool. A common meeting ground for all QA stakeholders, it enables full visibility into the testing process and a deeper broader understanding of testing results.

Features:

- A vast array of third-party integrations with common bug trackers, automation tools, and robust API for the rest.
- Fully customizable & flexible for the ever-changing needs of QA teams: customize fields, views, permissions, issue workflows and more
- Reuse tests and correlate results across different releases and products.
- Unique hierarchical filter trees – organize everything and find anything quickly.
- Never work twice – with anti-bug duplicates, permutations, step parameters and call to test
- Visualize data with advanced dashboards and reports
- Fast professional and methodological support

5) Zephyr Scale

Zephyr Scale is a scalable, performant test management solution inside Jira, with advanced test planning, reporting, and reusability features.

Features:

- Reuse tests and eliminate duplication: cross-project hierarchical test libraries, parameters, test data, and shared steps
- Improve visibility, data analysis, and collaboration: Access over 70 cross-project reports, 60 gadgets, and 60 Confluence macros
- Easier to audit and trace: Detailed change history, test case versioning, end-to-end traceability with Jira Issues and Confluence pages
- Test automation and DevOps ready: Empower teams with BDD, CI/CD and automation integration using the built-in, free REST API. Publish automated test-execution results from Jenkins, Bamboo, and other tools

6) SpiraTest

SpiraTest is a state-of-the-art Test Management solution for teams big and small. Embracing fully the agile way of working, Spiratest helps you manage requirements, plans, tests, bugs, tasks, and code in a single environment. SpiraTest works out-of-the-box with minimal configuration and conforms to your needs, methodology, workflows, and toolchain.

Features:

- Effortlessly generate tests from requirements, and bugs from tests
- Easily manage test cases, sets, and runs in manual and exploratory testing
- Create tests based on parameters with seamless end-to-end traceability throughout
- Manage your processes and teams through executive dashboards that display a top-down view of your projects, with visualization, and business analytics at your fingertips
- Comes as an on-premise and saas version.
- Integrates with over 60 apps

7) TestMonitor

TestMonitor is an end-to-end test management tool for every organization. A simple, intuitive approach to testing. Whether you're implementing enterprise software, need QA, building a quality app or just need a helping hand in your test project, TestMonitor has you covered.

Features

- Requirement and risk-based testing.
- Advanced test case design capable of supporting thousands of cases.
- Robust planning tools with multi-tester runs and milestone cloning.
- Comprehensive result tracking.
- Integrated issue management.
- Smart reporting with many filter and visualization options.
- Revolutionary simple UI.
- Third-party integrations featuring Jira, DevOps, and Slack. REST API included.
- Professional support with quick response time.

Automated Testing Tools

This category of tools helps automate functional and Regression Testing of your application under test.

8) Avo Assure

Avo Assure is a 100% no-code automation testing tool that enables you to test end-to-end business processes with a few clicks of the buttons. Being heterogeneous, it makes testing seamless across web, desktop, mobile, ERP applications, Mainframes, associated emulators, and more.

Features:

- Create and execute test cases without writing code
- Achieve E2E test automation and over 90% coverage
- Enable accessibility testing
- Define test plans and design test cases through the Mindmaps feature
- Integrate with tools like Jira, Jenkins, ALM, QTest, Salesforce, Sauce Labs, TFS, etc.
- Execute multiple scenarios through smart scheduling
- Easily interpret the intuitive reports

---

9) Kobiton

Kobiton empowers testing and development teams to automate Functional, Performance, Visual, and Compatibility testing across real mobile and IoT devices.

Solutions:

- Access to Real devices in a Public or Private Cloud
- “Cloudify” local devices for shared remote access
- On-premises solutions
- Scriptless/and or scripted automated Functional, Performance, Visual, and Compatibility Testing
- Support for leading open-source automation tooling and frameworks
- Generate 100% open-standard Appium code with every scriptless test
- AI-assisted remediation of Functional and Visual issues
- Unlimited user policy
- In-depth test session exploration
- Access and share analytics around app quality
- Access to real devices within your IDE for rapid debugging
-

10) Parasoft Continuous Quality Suite

Parasoft's suite of automated software testing tools integrates quality into the software delivery process for early prevention, detection, and remediation of defects. Parasoft continuous quality suite covers static code analysis, unit testing and code coverage, API testing and service virtualization, and UI testing, enabling delivery at speed and compliance with industry and security standards.

Features:

- AI-powered test creation and execution
- Low-code application testing
- Extensive dashboards for quality reporting and analysis
- Support for 120+ message formats and protocols
- Integration with CI/CD pipeline and Agile DevOps workflows