



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS730
Software Architectural Specification

Customer Care System

Student number: 11242796

Lecture(s):
Stacey Baror

1 Introduction

The purpose of this document is to provide a detailed architecture design of the customer care system by focusing on these quality attributes selected: Performance, Reliability, Scalability, Security, Maintainability, Cost, Usability. The document further provides a high-level overview and explains the architecture of the Customer Care System. Defines the goals of the architecture, the technologies supported by the system, architectural styles, and components that have been selected. The document provides a rationale for the architecture and design decisions made from the conceptual idea to its implementation.

1.1 Purpose

The Software Requirements Specification (architectural specification) provides a comprehensive architectural overview of the Customer Care System. It presents several different architectural views to depict the different aspects of the system. This document will address the background for this project, and the architecturally significant functional requirements. Each of the quality attributes will be described through a comprehensive set of scenarios followed by an architectural overview, which includes an overview and a full description of patterns and tactics that will be used to address the core quality attributes. This will be followed up by a look at a couple of views into the system.

1.2 Scope

The scope of this Software Requirements Specification (architectural specification) is to explain the architecture of the Customer Care System. This document describes the various aspects of the Customer Care System design that are considered to be architecturally significant. These elements and behaviors are fundamental for guiding the construction of the Customer Care System and for understanding this project as a whole. Stakeholders who require a technical understanding of the Customer Care System are encouraged to start by reading the Software Requirements Specification documents developed for this system.

1.3 Background

Customer Care System (CCS) allows a company to manage the relationships company has with its' customers, using a combination of people, processes, and technology. It helps a company to learn more about their customers and customer's behavior so they can develop stronger relationships that last longer with the customers. Key stability in today's dynamic marketplace is to attain and retain customers. As it is in the past, companies cannot hold in the market by only offering some product or service to their customers. In addition to these, companies must have a good relationship with their customers in order to have long-term relationships with them. Besides, while keeping existing customers, getting new customers is an essential issue in today's business world. Therefore, satisfying customer needs becomes a vital issue for companies. Customer Complaints are considered vital and significant information that can be utilized to attain customers' satisfaction. Consequently, establishing a customer care system is essential towards addressing customer dissatisfaction and preventing similar problems from reoccurring. To solve this problem, the Customer Relationship Management strategy must be adopted by companies. Customer Relationship Management is a company strategy that is essential for companies to reduce costs and increase efficiency while having good relationships with customers.

Customer Care System will be an oriented by Web-application which will be used by customers to make complaints about their dissatisfaction on provided services. This system will be able to handle complaints by recording and giving feedback for each raised complaint. Since the organization has to deal with several complaints, The Customer Care System will have a service for classifying the complaint, then automatically direct to the responsible department and the service for finding the similar complaint to avoid submitting the duplicate complaint. Thus, the benefits of saving time and labor. The Customer Care System will be able designed in such a way that it will be able to reduce the time and procedures for complaint handling, increase the channel for filing the complaint, and increase the channel for progress reporting and tracking the status of the complaint. The System will also be able to perform some semantic data analysis which can be a good reference to find out users' needs from an electronic complaint and the handling process of this complaint in the body of any organization.

2 Non-Functional Requirements

To have a properly designed and efficient system, non-functional requirements relating to 'how' the system should operate needs to be considered. Key non-functional requirements are listed and explained below.

Performance, Reliability, Scalability, Security, Maintainability, Cost and Usability

2.1.1 Quality requirement

2.1.1.1. Availability and Reliability

The Availability requirement addresses the below:

End-User Availability (complement of Planned Outages), Failure Rate or Unplanned Outages and Operational Continuity

End-User Availability (complement of Planned Outage)

- The end-user availability relates to the time that the system is expected to be available to the users.
- This requirement includes planned outages but not failures. Due to the nature of the customer care system process, the real-time system needs to be available to the users 24 hours x 7 days over 365 days.
- The target uptime of the system is 95 percent availability whilst the integration with the third party's component needs 95 percent availability. The task management dashboards, and the management platforms components need to be available 90 percent of the time.
- A planned outages window is also necessary for regular system maintenance and a backup server might be required during that time.

Failure Rate The failure

- Rate or unplanned outages relates to the reliability of the system and measures how often it is acceptable for the system to fail during the 24 hours x 7 days x 365 days period.
- Due to the nature of the customer care system, a minimal failure rate is essential for some of the components to minimize disruption to the service. The acceptable failure rate for the database and client communications of the system is 0.5 percent of the time whilst the integration with third parties' acceptable failure rate is 5 percent of the time.
- The acceptable failure rate for the task management dashboards and the management platforms is 10 percent of the time.
- The resource management component has an acceptable failure rate of 0.5 percent of the time. For instance, a UPS (uninterrupted power supply) can be used to minimize outages due to power failure or load shedding.

Operational Continuity

- Operational continuity relates to unplanned outages, system downtime and how quickly can the system recover from these failures.
- Due to the nature of the customer care system, any disruption to normal operations needs to be resolved quickly to restore the system to normal running conditions.
- Each component of the customer care system needs to have its recoverability requirements and based on those requirements the system needs to be able to recover easily from the failure of any of the components.
- During an unplanned outage, recovery time for the database and client communication of the system is sub-one hour whilst the integration with third parties acceptable recovery time sub-two-hour.
- The task management dashboard component needs to be recovered within 12 hours whilst for the management information platform component, the acceptable recovery time is 48 hours.

2.1.1.2 Scalability

The Scalability requirement addresses the need for the system to satisfy:

Deployment, Increase/Decrease in user volumes

Deployment

- Deployment relates to whether the solution will be deployed on a large or small scale.
- Firstly, the solution will be deployed on a small scale, this will be the staff and management users portal where the system will be used by a small number of company users. The system should not be cumbersome to administer.
- On a larger scale, the solution will be deployed for all users, included customer users where the number of customers using the system is higher.
- The activity level will also be significantly higher once a system is available for customers to use, due to the higher number of events that occur.
- Since all the customer care system resources are both for desktop/computer and mobile/phone use, a 'LITE' version of the solution might also be deployed to accommodate those resources.

Increase/Decrease in user Volumes

- An increase/Decrease in user volumes relates to the number of transactions that the system needs to be able to process.
- During major events such as Black Friday specials, festive season, easter season, or just new product offers, the system needs to anticipate an increase in processing due to a potentially larger volume of users during those events.
- The system needs to be able to satisfy the increases and decreases in processing without influencing system performance.

2.1.1.3 Performance

The Performance requirement addresses:

Latency time, Utilization

Latency time

- Latency time is the time required for the application to complete a task and present the information to the user.
- The user needs to be able to launch the application quickly and log into the application with minimal latency, sub 5 seconds.
- Also due to the nature of the customer care system, minimal latency of sub 5 seconds for on-screen navigation, information capture, and information upload and transfer is critical.
- The latency of the user interface should never exceed 5 seconds for these activities. For instance, the user should be able to navigate from one tab to the other within 2 seconds.
- Once the user and complaint details have been captured, the system should be able to submit the information to the database within 2 seconds.
- Retrieving information, for instance for a complaint history, also needs to be minimal.
- Sending notifications and also generating reports, the latency should also be within 20 seconds, as this service will require third-party processors such as mail server and reporting server.

Utilization

- Utilization is the maximum load of the components on the system, for instance, CPU, network, database.
- Utilization parameters need to be defined and monitored to ensure that the system performs at its best at all times.
- For instance, CPU usage needs to be monitored to ensure that system resources are used efficiently.

- The customer care system peak period is estimated to be 8 hours out of 24 hours and the system is expected to accommodate the load during those 8 hours and take appropriate corrective actions proactively to stay within utilization parameters.

2.1.1.4 Usability

The Usability requirement addresses:

User Interaction, System Navigation

User Interaction

- User Interaction is how easy it is for the user to interact with the application to perform required transactions.
- This requirement also covers learnability whereby the user needs to be able to use the application without extensive training, typically 2 or 3 days of training.

Application navigation

- Application navigation is how easy it is for the user to navigate within the application.
- The user needs to be able to access the various information they require in a quick, consistent, and intuitive way.
- For instance, drop-down lists and predictive text need to be available to identify product types and complaint status types.
- The user needs to be able to perform functions in the system via a menu or for more experienced users using short cuts commands or keys.

2.1.1.5 Maintainability

The Maintainability requirement addresses:

System Administration and Maintenance, Customization, Back-Up/Recovery/Restoration

This requirement is supported by the scalability as well as the availability and reliability requirements

System Administration and Maintenance

- System administration and maintenance is the need for the system to be easily maintainable with minimal disruption to the system.
- Both the core system and subsequent add-ons need to be easily maintainable.
- For instance, it should be easy for the system administrator to add, delete or update users' profiles.

Customization

- Customisation is the need for the application to be easily customizable if required.
- Functions can be added or modified with minimal disruption to the application.
- Fields can be added or moved, and reports created or reformatted if required.
- The client would like to keep the degree of customization to an optimum level in line with the maintenance and enhancement cost.

Back up/Recovery/Restoration

- Back up/Recovery/Restoration is the need for the system's functions and data to be backed up and readily restored when required.
- Back-ups are important to ensure that historical data is kept as per legal requirements.
- Also in the case of major disasters, it is essential that if the building could burn down, the system can be restored from the disaster recovery point and recover all data for operational continuity.
- As part of the disaster recovery plan, there will be another site in the metropolitan where the backups will be stored.
- Information from the customer care system will be stored at this site and will be accessible if need be.

2.1.1.6 Security

The Security requirement addresses:

Authentication, Authorisation, Information Security, Auditing, Security Management and System Administration

Authentication

- Authentication is the ability of the application to identify if a user is allowed to access the system.
- A Username and password combination-based authentication will be implemented.
- Passwords can be in the form of a string of letters, numbers, or special characters.
- To protect yourself, a user needs to create strong passwords that include a combination of all possible options.
- Before a user can access their account, they must prove that they own the correct login credentials.
- Services typically present a screen that asks for a username along with a password.
- Then, the system will compare the data inserted by the user with the values previously stored in an internal repository.

Authorization

- Authorisation is the ability of the system to control the functionalities that the user are authorized to use.
- Each user will have a sign on and each sign-on will have a list of roles associated with it.
- For each role, access control rules will be defined to restrict the user to the functions that they are authorized to perform under that role.
- For instance, Customer users cannot see the staff and management portal since this is an internal portal for resolving all those complaints raised.

Information Security

- Being a solution that deals with the customer's personal information, the solution needs to comply with the organization's security standards and with the industry's best practice.
- Only authorized users should be allowed to access the information.
- Relevant national guidelines and legislation need to be adhered to to ensure secure transmission and storage of data.
- The organization should comply with POPI acts, the customer care system will be built under the compliance of the organization towards POPI Security Safeguards.

Auditability

- Auditing is the ability of the application to track changes performed by users.
- The system needs to keep a log of activities performed by the users for instance capturing information about a complaint, augmenting the complaint information with additional information, etc.
- The log needs to include a timestamp and user identity. This forms the basis for an audit trail of each complaint.
- Further information is available in the Software Requirement Specification document.
- The information can also be used for system usage trends if required. It becomes a key part of providing information for a report or complaint.

Security Management and System Administration

- The system needs to allow users with system administration rights to remotely log in to perform system maintenance and configuration if required.
- Users with system administration rights need to be able to define new users and their roles, update access to functions, etc remotely.
- The support of the system could be outsourced. In such an event the system should provide the features that will facilitate this.

2.1.1.7 Cost

These business goals about the reduction of costs associated with the system. All types of costs are included in this category. As a result, this category includes all of the goals in other cost-related categories, in addition to other goals:

- Use Commercial-off-the-shelf components whenever they are reliable enough.
- Adopt a build rather than buy approach to software.
- Customer care system will lower the cost and cycle time for system integrators.
- Customer care system's goal is to enable system builders to integrate systems on time and within the cost while meeting performance needs. The number one objective is to drive down the cost of the system.
- Automate operations to minimize operational costs.
- Improve Obsolescence Management.
- Support technology refresh.
- Reduce and eventually eliminate the need for overtime to support the setup and operation of the system.
- System A must support two kinds of debugging: 1. scenario debugging and 2. software debugging. The system must be responsive to customers. It must take less time to implement fixes and to add new features for analysis, test, and integration customers. Maintenance must be easy and quick.
- The cost of ongoing software maintenance is minimized. Latent defects are corrected without major effort.
- The system must have an operational availability of 95 percent and a maintenance ratio that does not exceed 0.05 maintenance man-hours/operating hours.
- Lower maintenance and support costs. There are over 10,000 systems fielded and several software baselines. The customer care system will be in the field for the foreseeable future.
- Reduce lifecycle costs.
- Reduce manning.
- Minimize sustainment costs.

2.1.2 Quantification of the quality requirementt

2.1.2.1 Availability and Reliability

- The system defect rate shall be less than 1 failure per 1000 hours of operation.
- The system Percentage of uptime (Minimum of 95 percent) Less than 20 seconds shall be needed to restart the system after a failure 95 percent of the time.
- No more than 1 per 100000 transactions shall result in a failure requiring a system restart.
- Network failure detected and recovered in less than 2 minutes Restart after disk failure in less than 5 minutes.

2.1.2.2 Scalability

- The Home page supporting 5 thousand users per hour must provide 5 seconds or less response time in a Chrome desktop browser, including the rendering of text and images, over an LTE connection.
- The effort needed to administer the customer care system (as measured in hours per month of system administrators' time) shall not increase with an increase in the number of users.
- Table below shows number of active users and active transactions that customer care system will accomodate per second.

Measurement	Threshold
Number of concurrent users	100
Number of daily visitors	10000
Number of New Complaints	10000

2.1.2.3 Performance

- Production of a data report shall take less than 20 seconds for 95 percent of the cases.
- The system shall be able to process 100 complaint transactions per second in peak load.
- When connecting to the server the delay should be in less than 20 seconds for sake of good communication.
- Below is the expected time customer care system is required to uphold when performing selected these tasks

Measurement	Threshold
Launch the application	2 seconds
Screen navigation	5 seconds
Submit the information	2 seconds
Retrieving information	5 seconds
Sending notifications	20 seconds
Generating Reports	20 seconds

2.1.2.4 Usability

- Customer care system users shall perform tasks in 5 minutes, Experienced users shall perform tasks in 2 minutes.
- Customers with technology experience from other systems: In their first attempt, they must be able to capture and submit a complaint within an average of 2 minutes.
- Customers without previous customer care systems experience: In their first attempt, 90 percent of them must be able to capture and submit a complaint within 4 minutes.
- Customers with at least 3 complaint over at least a month: They must be able to capture and submit a complaint with an average of 30 seconds

2.1.2.5 Maintainability

- The product shall provide facilities for tracing any database field to places where it is used.
- Updates of a new version shall leave all database contents and all personal settings unchanged.
- The cyclomatic complexity of code must not exceed 7.
- No method in any object may exceed 200 lines of code.

2.1.2.6 Security

- At least 99 percent of intrusions shall be detected within 10 seconds by using Network-based Intrusion Detection System (NIDS).
- NIDS can be easily introduced into an existing network with minimal disruptions.
- Host-based Intrusion Detection System (HIDS) will also be implemented to detect intrusions.
- HIDS monitors system data and looks for malicious activity on an individual host. HIDS can take snapshots, and if they change over time maliciously, an alert is raised.
- HIDS can access encrypted data packets and can detect attacks with elusive capabilities.
- The system shall identify all of its client applications before allowing them to use its capabilities using Server-Side Device Detection With JavaScripts.
- Customer database authorization works 99.9 percent of the time this will be ensured by Regularly updating the operating system and patches.
- Regularly testing your database security by using 3rd party services that specializes in penetration testing.
- regularly backup of database will be performed to ensure that any backups are encrypted and stored separately from the decryption keys
- Encrypted backups will not be stored alongside decryption keys in plain-text.
- Regularly backing up customer care system not only to protect against hackers but other failures as well, such as problems with physical hardware
- e-mail encryptions are secure 99.9 percent of the time by using PGP: Pretty Good Privacy, a hybrid approach that compresses the text, this does not only save disk space but increases security.
- PGP creates a session key, which is a one-time use secret key. The text is encrypted, including the session key with it; the public encrypted session key is sent along with the coded/encrypted text

3 Architectural Design

Software system design consists of the activities needed to specify a solution to one or more problems, such that a balance in fulfillment of the requirements is achieved. A software architecture design method implies the definition of two things. First, a process or procedure for going about the included tasks. Second, a description of the results or type of results to be reached when employing the method. From the software architecture point-of-view, the first of the aforementioned two, includes the activities of specifying the components and their interfaces, the relationships between components, and making design decisions, and document the results to be used in detail design and implementation. The second is concerned with the definition of the results, i.e. what is a component and how is it described, etc

Software architecture is the highest abstraction level at which we construct and design software systems. The software architecture sets the boundaries for the quality levels resulting systems can achieve. Consequently, software architecture represents an early opportunity to design for software quality requirements, e.g. reusability, performance, safety, and reliability. The design method must in its process have an activity to determine if the design result, in this case, the software architecture, has fulfilled the requirements. We only consider design methods with such activity as considered complete.

3.1 Architectural patterns

An architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture within a given context. Architectural patterns are similar to software design patterns but have a broader scope.

Customer Care System will be based on Multi-Tier architecture design, The multi Tier system architecture shows the overall organization and communication between the users and the system. As shown in Figure below, the web-based Customer Care System will be divided into 4 layers; Presentation layer, Application layer, domain layer, and data source layer.

Maintainability quality attribute of any software system is an important design quality. It plays an important role especially when a system is built using open source technologies.

Customer care system design solution kept **"Maintainability"** architecture quality attribute as prime focus.

"Maintainability" is the first order of precedence for quality attributes that the architecture must address and then proceed with the design of the system.

I have chosen Multi-Tier Architecture which addresses my two key target quality attributes: Maintainability and Usability. There is little more effort involved in building the application module but it helps to maintain the components decoupled so that in future upgrades and enhancements can be done independently and this option does not disturb existing user experience.

The aim of using the Multi-Tier Architecture is not just to make it easier only when you change your database engine or programming language. It is also useful when you want to either replace your Presentation layer or create an additional Presentation layer. Another not-so-obvious benefit that can only come from actual exposure to having developed multiple applications using the Multi-Tier Architecture is that it becomes possible to create a framework for building new applications around this architecture.

As each of the layers specializes in just one area of the application it is possible to have more reusable components which deal with each of those areas. Such components can either be pre-built and delivered as part of the framework, or generated by the framework itself. This reduces the amount of effort needed to create a new application and also reduces the amount of effort needed to maintain the system. the Customer care system's non-functional requirement for Multi-Tier Architecture is based on:

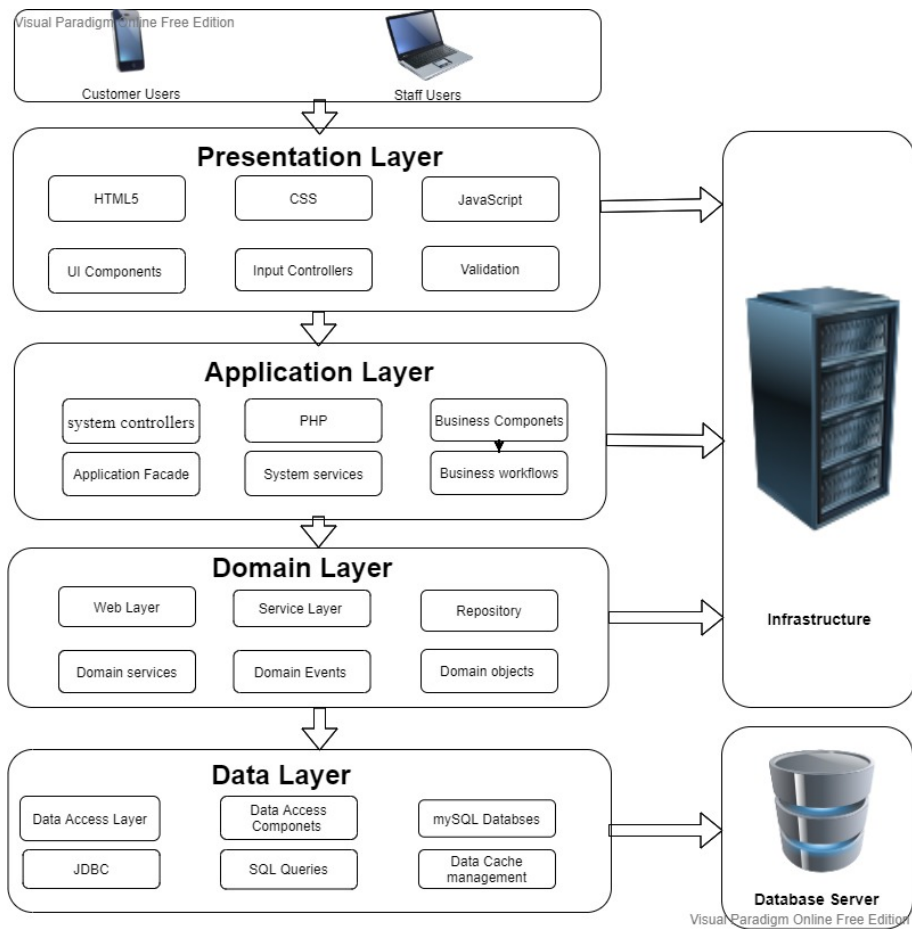
Maintainability

- Changes to the components in one layer should not affect any others layers.
- Also, if different layers require different skills (such as HTML, CSS, and JavaScript in the presentation layer, PHP and Python in the business layer, MySQL in the data access layer) then these can be managed by independent teams with skills in those specific areas.
- Each tier is independent of the other tiers, which allows changes and patches to be performed on a part of the system without affecting the application as a whole.
- Maintainability is an extremely important consideration in any software project, so this is a large benefit.

Usability

- Separating the application into multiple layers makes it easier to implement reusable components.
- A single component in the business layer, for example, maybe accessed by multiple components in the presentation layer, or even by several different presentation layers (such as desktop and the web) at the same time.

CUSTOMER CARE SYSTEM MULTI-TIER ARCHITECTURE DIAGRAM



Presentation Layer

- It consists of a web-based user interface, The presentation layer will provide the user with a graphical interface for interacting with the system.
- It will be composed of HTML, CSS, JavaScript, and other related files that will run in the user's web browser of choice.
- The main goal of this layer is to provide everything the user needs to complete their tasks in the system.
- The quality attributes that pertain to usability and frontend design will have the biggest pull on the presentation.
- Through the Presentation layer components, Customer care users can register their complaints to the organization and track the status of their complaints.
- The management users and staff users can also provide an immediate response to their customer's complaints through web Presentation layer components.
- Individual components in the proposed architecture provide different functionalities.

Application Layer

- It consists of the core of the system. The complaint handling and feedback components.
- The role of the application logic layer is to encapsulate the system controllers, which implement much of the core business logic.
- It will also serve as the connection between the presentation layer and the domain layer thus maintaining the separation of concerns.
- The application layer is the heart of the application. In this layer, information collected in the presentation layer is processed - sometimes against other information in the data layer - using business logic, a specific set of business rules.
- The application layer can also add, delete or modify data in the data layer.
- The application layer creates a connection between the web presentation layer and data layer by using network protocols such as HTTP.
- Web servers accept the request from both users and complaint handlers and redirect it into a database and vice versa.
- It is responsible for any kind of communication among system users, complaint handlers, and database.

Domain Layer

- The domain layer contains all of the system's object representations of data in the system.
- This also includes associated methods for any objects that contain their functionality.
- The service layer is in charge of modularizing different features (services) of the application.
- Certain features, such as reporting, require access to external APIs, the service layer will be in charge of interacting with those external APIs.
- An individual service can also use the application logic layer to perform core business logic.
- Each service will act as a facade, serving as a general interface for a feature that can be accessed by the presentation layer.
- These facades (interfaces) will provide the ability to manipulate or fix the code underneath each service without affecting how that service is called by other layers.
- Modularizing the services within the system will also make it easier to pinpoint potential defects, allowing for easy maintainability.
- Each service can be tested individually, which allows for good systemwide testability.

Data Layer

- The data access layer contains all of the mappers to the data in the system.
- The mappers are in charge of the coordination of all communication between the objects in the domain layer and their corresponding tables in the database.
- This ensures that domain layer objects do not know the database, its schema, or any SQL interface.
- All persistent information and any external API integration (e.g. SIS, Simplicity) make up the data layer.
- This includes the MySQL database that will contain all of the data for the system.
- At this time there is no expectation for integration with external systems, but the system should be architected to accommodate such integrations in the future.
- This layer is responsible for the management of the database, retrieving, updating, and storing data.

- The application layer uses web server components and network components so that data is transferred to the presentation layer and presented to the users and complaint handlers.

Relations between Layers:

Presentation Layer to Application Layer

- The presentation layer represents the view of the system, and the application logic layer contains the controllers, which house the logic for the different roles.
- In following with the MVC architectural pattern, the controllers take information from the view and use it to modify or request related data in the model.
- This prevents the view from directly modifying the model, and instead has the view display changes to the data.

Application Layer to Domain Layer

- The application layer modifies the data encapsulated in the domain layer according to the business logic rules before it reports changes back to the presentation layer to be displayed to the user.

Domain Layer to Data Layer

- The domain layer is the active representation of the information stored in the data layer.
- When it is time to store data according to the procedures defined in the upcoming detailed design stage, the changes to the domain layer will be pushed to the data source layer to create a persistent copy of the data to be stored across sessions through the data access layer.
- All interactions between the data access layer and Data Source Layer will indirectly be SQL queries.

3.2 Architectural Styles (tactics)

Component-based architecture Architectural Style:

- Customer care system will be based on a Component-based architecture style.
- A component-based architecture focuses on the decomposition of the design into individual functional or logical components that represent well-defined communication interfaces containing methods, events, and properties.
- It provides a higher level of abstraction and divides the problem into sub-problems, each associated with component partitions.
- The primary objective of component-based architecture is to ensure component reusability. A component encapsulates the functionality and behaviors of a software element into a reusable and self-deployable binary unit.
- For customer care systems there are standard component frameworks used such as UI components, web services, application servers, and database servers.
- This will help Reduced time in the market and the development cost by reusing existing components and Increased reliability with the reuse of the existing components.

Deployment

- As new compatible versions become available, it is easier to replace existing versions with no impact on the other components or the system as a whole.
- Components implement well-known interfaces to provide defined functionality, allowing development without impacting other parts of the system.

Batch Processing

- The user inputs their information through forms displayed in their web browser of choice.
- The client then sends that data to the server through the service layer to the application logic layer for processing, manipulation, and transformation before being written to the domain layer.
- The information may be forwarded through the data access layer to the data source layer to be written out to the database for more permanent persistence through the use of CRUD operations.

- Data may also be retrieved from the data source and manipulated by the application layer before being displayed on the view to the user.
- Of course, any transformation of data only occurs as necessary, and there may be a case where the unformatted data is called for, such as displaying all the information in a given table.

Security

- Customer care system must support authentication for both remote and local users for access to various capabilities.
- Customer care system must support confidentiality as the system might have multiple users with different levels of permission simultaneously using the system.
- Physical security is not an issue, since the system is protected using a door lock.
- What is of concern is the protection of personal data especially against viewing by service technicians and protection against illegal manipulation of the system.
- In the event of an accident, product liability is a concern, and illegally manipulating the system may be a cause of an accident.

3.3 Architectural Constraints

The Customer care system is designed under the following constraints:

USER CONSTRAINTS:

- Customer care system will be presented in english only.
- There should be no restrictions imposed by the software on users. Only policy decisions (permissions, access privileges, etc.) should prevent any user from accessing any part of the Customer care system from any local or remote station.
- Similar functionality should be presented to the users using similar user interfaces. However, user interfaces should reflect access modes and operating levels.

SOFTWARE CONSTRAINTS:

- The customer care system is to be developed using standard methodologies and development environments. One of the goals of the Customer care system is that all components be easily (preferably automatically) combined into an integrated system.
- Customer care system developers should maintain accurate change logs showing software modifications as they are applied to the system software.
- Customer care system developers should adhere to a standard method for the reporting and recording of errors from both internal and external sources.
- Customer care system subsystem packages must provide modules for the testing and diagnosis of the subsystem.
- The customer care system must be version labeled, both in source and binary form. The version information is to be retrievable from executing software via control commands.
- The use of a multi-tiered architecture design in the Customer care system means that the integrated system can be tested and developed independently of the target hardware. This is useful in maintenance and testing. Therefore all hardware subsystems must provide a software simulation module that responds reasonably to commands directed at that hardware.
- The customer care system will be built with PHP version 8 an open-source web scripting language and python 3.5 an open-source scripting language for web applications.
- Data will be stored in a MySQL database, also open source. The user interface will be developed in HTML5, CSS4, and JavaScript.

- `system()` or `popen()` from PHP will be used to call the Python scripts will be employing components from standard and commonly accepted libraries such as PHP Symfony and jQuery for JavaScript.
- PHP dependencies will be managed via the Composer package manager.
- The system will be able to run on any web server that supports PHP 8 and has a MySQL database, Git version control, free and open source version control system.
- The customer care system will only allow media files such as pictures. .PNG, .JPEG formats, and document with .PDF and .DOCX to be uploaded as part of complaint raised.
- Media files and documents must be less than 500 MB.
- Customer care system will run on Supported Desktop /Mobile Browser Versions Google Chrome versions 49+, Edge version 20+, Firefox version 48+, Safari IOS 11+.
- Customer Care does not support beta versions of Operating Systems or Web Browsers.
- 1024x768 resolution or higher is required for the internal web portal and 800x600 resolution or higher is required for the customer users access portal.
- Desktop minimum requirements to run customer cares systems: must meet the following minimum hardware requirements: Dual Core 1.6GHz or faster with RAM: 1 gigabyte (GB) (32 bit) or 2 GB (64 bit).

3.4 Actor-System Interaction models

Actor-system interaction modeling is modeling and design of how the customer care system interacts with the actors to carry out the use cases. Actor-system interaction modeling is accomplished by constructing a two-column table that describes, for each interaction, the actor input and actor action, and the system response.

Actor 1

<i>Precondition: This use case assumes that the customer user has logged into the system and has shown the home page.</i>	
Actor: Customer User	System: CCS
<ol style="list-style-type: none">1. TUCBW the customer user clicks the Register Complain button from the main menu.3. The Customer user fills in the complaint form with all required details and clicks "submit".5. TUCEW the customer user clicks the OK button to confirm the submission and clear the screen and return to the home page.	<ol style="list-style-type: none">0. The system displays the home page.2. The System displays the complaint form.4. The System validates and checks if all required information is supplied.
<i>Postcondition: the complaint is submitted to the database and assigned to the relevant management user immediately</i>	

Actor2

<i>Precondition: This use case assumes that the management user has logged into the system and has shown the management dashboard page.</i>	
Actor: management User	System: CCS
<ol style="list-style-type: none"> 1. TUCBW the management user clicks the new Complaints button from the main menu. 3. The management user selects new complaints to assigns to the staff users and clicks "submit". 5. TUCEW the management user clicks the OK button to confirm the submission and clear the screen and return to the home page. 	<ol style="list-style-type: none"> 0. The system displays the management dashboard page. 2. The System display all new complaint lodged. 4. The System validates and check if all required information is supplied.
<i>Postcondition: the complaint is assigned to the relevant staff user immediately</i>	

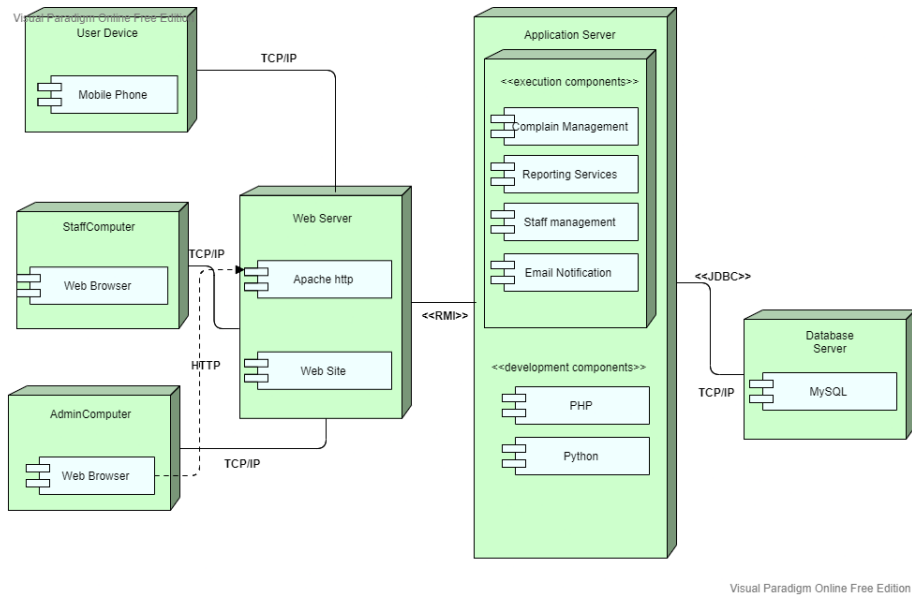
Actor3

<i>Precondition: This use case assumes that the staff user has logged into the system and has shown the task management page.</i>	
Actor: staff User	System: CCS
<ol style="list-style-type: none"> 1. TUCBW the staff user clicks the opened complaints button from the main menu. 3. The staff user selects one opened complaint from the list 5. The Staff user update complaint details by providing a reply and updating a complaint status to "closed" then clicks on submit button 7. TUCEW the staff user clicks the OK button to confirm the submission and clear the screen and return to the home page 	<ol style="list-style-type: none"> 0. The system displays the home task management page. 2. The System displays all opened complaints assigned to the staff user form. 4. The system displays the complaint details. 6. The System validates and check if all required information is supplied.
<i>Postcondition: the resolved complaint is sent to the customer in form of an email reply</i>	

3.5 Deployment Model

- The deployment model of the Customer care system is based on the Multi-Tier architecture model.
- To describe the Customer care system, a deployment model shows the hardware components which are nodes i.e. client device, Web server, application server, and a database server.
- Further describe what software components run on each node and how the different pieces are connected, i.e. JDBC, RMI
- The system has 4 main components: client device, Web Server, application server, and database server. Users can access the system using either a mobile device or a computer through a Web server.
- The Web Server component defines users of the system who access the system through HTTP protocol.
- The web server sends a request using browser software found on a client machine and the system responses for the request reach the system through the Internet.
- The communication between the Web server and application server is through HTTP protocol.
- The third component of the system is an application server on which an application runs and communicates with the database to respond to the user.
- The fourth component of the system is a database used for providing responses and storing the data of the system.

CUSTOMER CARE SYSTEM DEPLOYMENT MODEL DIAGRAM



3.6 Technology requirements (technical)

The software engineering community has evolved enormously. Improved software development tactics, changing industry dynamics, changing customer needs, automated processes, etc. are some of the critical changes that have taken place over the past few years. So, for a software engineer, the primary challenge lies in staying updated with the current trends and being the best among the many professionals thriving in the industry.

Here are various technologies used in software engineering today:

Databases/Data storages:

MySQL, PostgreSQL, Oracle, MS SQL Server, Derby, Lucene/SOLR/Elastic-Search, MongoDB, LDAP, etc.

IDE: Eclipse, Idea, MS Visual Studio, Aptana Studio, XCode

Programming languages:

Java, PHP, Ruby, c#, C / C++, SQL, PL/SQL, ASP .NET, Objective-C, Visual Basic

Technologies, libraries, and frameworks:

- **Java:** JEE, Spring, Spring Security, Spring Boot, JPA, Hibernate, Flyway, Swing, JSF, Apache Wicket, Velocity, GWT, JXL, Oracle Portal
- **PHP:** CodeIgniter, Laravel, Symphony, Zend, Yii, WordPress, Joomla, Drupal, Magento, Prestashop
- **C#:** Microsoft Visual Studio, Telerik Framework, Entity Framework, ASP Web API, WinForms, ASP.NET MVC, Neodynamic SDK .NET:ASP.NET, Visual Basic

hosting services:

GitHub, Codenvy , Bitbucket

Testing tools and frameworks

Selenium, Protractor, Ranorex, JMeter, JBehave, Testing Anywhere, WebUI Test Studio, TestComplete, SOAP UI, EasyMock, Mockito, jMock, WebUI Test Tool, TCMS, Cucumber, ProjectLocker

Issue management tools

JIRA, Redmine, Rally, Trac, ActiveCollab, Rational ClearQuest, Bugzilla, Bugzero, Remedy, TestLink, TestTrack, etc.

Application and web servers:

Apache Tomcat, JBoss AS, Jetty, IBM WebSphere, Oracle Application Server, WebLogic, Windows Server IIS, Nginx

Technologies and protocols for implementing a Customer care system.

Customer care system will use Frontend technologies, reasons for choosing these technologies are mentioned below.

1. HTML5

- HTML5 will be used to create the semantic markup for the website. This creates the structure of the website in a DOM-like structure.
- semantic tags and js functions which improve development and improve web performance and functionality.
- Cost-effective Multi-Platform Development -A single batch of code can be used across platforms, devices, and different markets, this is an advantage because it results in lower development and maintenance costs over the website or web application's lifetime, enabling you to use your resources elsewhere.
- A better user experience - HTML5 offers a wider range of design and presentation tools across media types, giving the developers greater scope to produce better websites and web applications.
- This is vital from a business point of view, as user engagement and retention are key to increased site and system use and conversion.
- Creating an accessible and usable site or system means that users will be more likely to engage.

2. CSS

- CSS - it is used to style the page, basically add color, fonts, width, padding, etc.
- CSS saves time – You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- Easy maintenance – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Global web standards – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.
- Platform Independence – The Script offers consistent platform independence and can support the latest browsers as well.

3. JavaScript

- The logic part of the webpage which takes care of the button interactions, saves data and interacts with the server. Will be used to make the dynamic web pages like on click event, sliding pages, sliders.
- Encrypt: Use HTTPS/SSL to encrypt data exchanged between the client and the server.
- Set secure cookies: To ensure SSL/HTTPS is in use, set your cookies as "secure," which limits the use of your application's cookies to only secure web pages.
- Less server interaction – The customer care system can validate user input before sending the page off to the server. This saves server traffic, which means less load on the server.
- Immediate feedback to the visitors – Users will not have to wait for a page reload to see if they have forgotten to enter something.
- Increased interactivity – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- Richer interfaces – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to customer care system visitors.

4. PHP

- Open Source: PHP is open-source and free of cost, which helps developers to install it quickly and readily available for use.
- Platform Independent : PHP is mainly supported by all the operating systems like Windows, Unix, Linux, etc. The PHP-based developed web applications can be easily run on any platform.
- Simple and Easy: This advantage of PHP is simple and easy to learn and code. It is mainly organized code and clean, which helps the new developers also.
- Database - PHP is easily connected with the database and makes the connection securely with databases. It has a built-in module that is used to connect to the database easily.
- Maintenance - PHP framework is mainly used to make web application development easier and maintain the code automatically.

5.PYTHON

- Python is open-source, which means it's free and uses a community-based development model. Python is designed to run on Windows and Linux environments.
- Also, it can easily be ported to multiple platforms. There are many open-source Python libraries such as Data manipulation, Data Visualization, Statistics, Mathematics, Machine Learning, and Natural Language Processing
- Text classification is one of the most important tasks in Natural Language Processing. It is the process of classifying text strings or documents into different categories, depending upon the contents of the strings.
- Use Sentiment Analysis With Python to Classify product Reviews
- Python allows the use of natural language processing (NLP) techniques
- Python allows the use of machine learning to determine the sentiment of text
- Python allows the use of spaCy to build an NLP pipeline that feeds into a sentiment analysis classifier.

6.MYSQL

- Data Security - MySQL is globally renowned for being the most secure and reliable database management system used in popular web applications like WordPress, Drupal, Joomla, Facebook, and Twitter.
- On-Demand Scalability - MySQL offers unmatched scalability to facilitate the management of deeply embedded apps using a smaller footprint even in massive warehouses that stack terabytes of data.
- High Performance MySQL features a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for a flawless performance.
- The Flexibility Of Open Source - All the fears and worries that arise in an open-source solution can be brought to an end with My SQL's round-the-clock support and enterprise indemnification.
- Round-The-Clock Uptime MySQL comes with the assurance of 24X7 uptime and offers a wide range of high availability solutions like specialized cluster servers and master/slave replication configurations.