Project Design Phase
Problem–SolutionFitTemplate

Date	07NOV2025
TeamID	NM2025TMID06118
ProjectName	Garage Management System
MaximumMarks	4Marks

DataFlowDiagrams:

A**Data FlowDiagram(DFD)** is a visual representation of how information moves through a system.Itillustratestheflowofdatabetweenexternalentities,processes,anddatastores,helping developers and stakeholdersunderstand the system’s functionality at a glance.

Intheproject*“GarageManagementSystem”* ,theDFDdemonstrateshowservicerequests, vehicle details, and billinginformation movethrough various stages of the system.

The diagram shows how **customers** submit service requests, whicharerecorded in the **service database**.The**manager**assignsmechanicsfor eachjob,andoncetheworkiscompleted, **billing details** are generated and stored. Finally, **notifications** are sent to customersabout service completion and payments.

ThisDFDhelpsensureclearunderstandingofhowvehicleservicing,inventorymanagement,and billing operations are integrated within the system.

Example(Context-LevelDFD):

- **ExternalEntities:**Customer,Manager,Mechanic,BillingDepartment
- **Processes:**CustomerRegistration,JobAssignment,ServiceUpdate,PaymentProcessing
- **DataStores:**CustomerDatabase,VehicleDetails,ServiceRecords,BillingInformation

UserStories:

User stories describe what different users expect from the system in simple, goal-oriented language.Inthisproject,theycaptureessentialfunctionalitiessuchasvehicleregistration,job management, and billing — ensuring smooth garage operationsand customer satisfaction.

UserType	Functional Requirement (Epic)	User Story Number	UserStory/Task	Acceptance Criteria	Priority	Release

UserType	Functional Requirement (Epic)	User Story Number	UserStory/Task	Acceptance Criteria	Priority	Release
Customer	Vehicle Registration	USN-1	Asacustomer,Iwant toregistermyvehicle and service request online so that I can easilybookarepairor maintenance service.	Thesystemshould allowcustomersto register their vehicle with detailsandchoose a service type.	High	Sprint-1
Garage Manager	Job Assignment	USN-2	As a manager, I want to assign service jobs tomechanicsbasedon availability and skill.	Thesystemshould list available mechanics and assign them to pending service jobs.	High	Sprint-1
Mechanic	ServiceUpdate	USN-3	Asamechanic,Iwant to update the service status (in progress, completed)sothatthe managerandcustomer are informed.	Thesystemshould allow mechanics to update the job statusinreal-time.	Medium	Sprint-2
Billing Staff	Payment& Invoice	USN-4	As a billing staff, I want to generate service bills automatically based on job details.	Thesystemshould calculate service costsandgenerate a printable invoice.	High	Sprint-2
Admin	Reporting	USN-5	Asanadmin,I wantto view reports on completed services, pending jobs, and overallrevenue.	Thesystemshould generatedailyand monthly reports with summary statistics.	Medium	Sprint-3

FunctionalRequirements:

Followingarethe**functionalrequirements**oftheproposed*GarageManagementSystem* .

FR No.	Functional Requirement (Epic)	SubRequirement(Story/Sub-Task)
FR-1	CustomerRegistration	Customerscanregister withname,contact,andvehicledetails.
FR-2	Vehicle Information Management	Systemallowsstoringvehicledetailssuchasmodel,number , and service history.

FR-3	ServiceBooking	Customerscanbookaserviceappointmentthroughtheportal or in person.
FR-4	JobAssignment	Admin/Managercanassignjobstoavailablemechanicsbased on specialization.
FR-5	InventoryManagement	Trackavailablespareparts,updatestockaftereachservice,and generate purchase alerts.
FR-6	Billing&Payment	Automaticallycalculatecharges,generateinvoices,andrecord payments.
FR-7	ServiceStatusUpdate	Mechanicscanupdatetheprogressofongoingjobs.
FR-8	Notifications	SystemsendsSMS/emailupdatesonservicecompletionand pending payments.

FR No.	Functional Requirement (Epic)	SubRequirement(Story/Sub-Task)
FR-9	Feedback&Reporting	Customerscanprovidefeedback,andthemanagercangenerate daily/monthly reports.

Non-FunctionalRequirements:

Followingarethe**non-functionalrequirements**ofthe*GarageManagementSystem* .

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Thesysteminterfaceshouldbeuser-friendlyforbothstaffand customers.
NFR-2	Security	Onlyauthorizedpersonnelcanaccessjobassignments, payments, and reports.
NFR-3	Reliability	Thesystemmustaccuratelymaintainservicerecordsandjob histories.
NFR-4	Performance	Systemshouldhandlemultiplecustomerandservicerequests simultaneously without delay.
NFR-5	Availability	Theapplicationshouldbeaccessible24/7forbookingand status tracking.
NFR-6	Scalability	Thesystemshouldsupporttheadditionofnewservicetypes, users, and garages as it grows.
NFR-7	Maintainability	Thesoftwareshouldbeeasytoupdateandmaintainwith minimal downtime.
NFR-8	DataBackup	Allrecordsshouldbebackedupdailytopreventdataloss.

TechnicalArchitecture:

Theproposed*GarageManagementSystem* willbedevelopedasa **web-basedapplication**that helps managecustomerregistrations, vehicleservicetracking, mechanicjob assignments, and billing in an efficient and digital manner.

Thearchitectureincludes**threemajorlayers**—

1. **PresentationLayer(Frontend)**foruserinteraction,

2. **ApplicationLayer(Backend)**forbusinesslogicanddataprocessing,and
3. **DatabaseLayer(Storage)**formanagingdatasecurely.

ExternalAPIssuchaspaymentgatewaysandnotificationservicesareintegratedtoenhance functionality. The systemis deployed on a cloud-basedinfrastructure for scalability and accessibility.

Example: Centralizedgaragemanagementplatformaccessibletocustomers,managers,and mechanics via web and mobile devices.

Reference:
https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

S.No	Component	Description	Technology
1	**UserInterface**	Customers, managers, and mechanics interactthrougharesponsivewebportal.	HTML5,CSS3, Bootstrap5,JavaScript
2	**ApplicationLogic–1**	Handlescustomerregistrationandservice booking workflows.	Node.js/Express.js
3	**ApplicationLogic–2**	Assignsservicejobstomechanicsand tracks job progress.	RESTfulAPIs
4	**ApplicationLogic–3**	Generatesautomatedinvoicesandsends status notifications.	Python(Flask)/Twilio API
5	**Database**	Storesdetailsofcustomers,vehicles, services, and billing records.	MySQL/PostgreSQL
6	**CloudDatabase**	Cloud-hosteddatabaseforhighavailability and data backup.	AWSRDS/Firebase
7	**FileStorage**	Storesservicereceipts,reports,and customer feedback files.	AWSS3/Cloud Storage
8	**ExternalAPI–1**	SMSandemailnotificationintegrationfor service updates.	Twilio/SendGridAPI
9	**ExternalAPI–2**	Paymentgatewayforonlinebillpayments.	Razorpay/PayPalAPI
10	**MachineLearning Model**	Predictive maintenance suggestion (optional future enhancement).	TensorFlow/Scikit-learn
11	**Infrastructure (Server/Cloud)**	Hostedandmanagedonscalablecloud services.	AWSEC2/Google Cloud Platform

Table–2:ApplicationCharacteristics

S.No	Characteristics	Description	Technology
1	**Open-Source Frameworks**	Usesopen-sourceframeworksfor flexibilityandcost-effectiveness.	Node.js,Bootstrap,React
2	**Security Implementations**	Implementsrole-basedaccesscontroland encrypted data storage.	JWT Authentication, HTTPS
3	**ScalableArchitecture**	Easilyexpandableformultiplegarage branches and users.	CloudLoadBalancing , Microservices
4	**Availability**	Systemhostedonacloudserverensures 24/7 uptime.	AWSCloud/Azure
5	**Performance**	OptimizeddatabasequeriesandAPI caching for faster response.	Redis/IndexedDB Queries
6	**Maintainability**	Modularstructureenableseasyupdates and maintenance.	MVCFramework (Express/ React)
7	**Integration**	Supportsthird-partyAPIsforpayments and communication.	REST/JSONAPIs