



## EXERCISE 2

### Question 1.

Personio helps improve operational excellence to pave the way for strategic work to shine. That means accelerating daily work, reducing administrative chaos, and ensuring that every core HR process works like a dream.

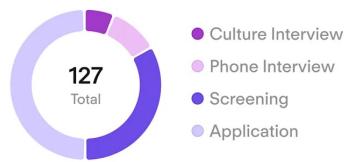
Their market revolves around small and mid sized companies

What functionality does its software offer to its clients?

As an AI Software Company Personio offers following functionalities to its clients →

- ① Hire and retain Exceptional Talent → *Saves recruiting Costs*

Application Funnel



• Speeds up ramp time with guided Onboarding

- ② Influence Strategic Decisions →

Gender



Salaries



Headcount



Creates easy to use reports on diversity of workforce, retention, Salary Progression and other key signals for smart decisions

### (3) Automate Tasks so the Company can Scale



Use templates to create automations for important HR events.

Create and send documents and provide a way for employees to self serve information so the company can scale.

### (4) Create Seamless Data Transfer

Build seamless flows of real time, accurate data across all your tech systems by connecting



to 200+ other trusted workplace platforms your employees are already using.

- Recruiting, On-Offboarding, Absence Management, Time tracking, Performance & Development, Payroll, Reporting & Analytics, Compliance, Employee Self Service

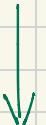
↓  
Manage employee info  
apply for leaves and access documents.

It's Cloud based Service

User Interface Layer

- Web / Mobile App

(HR Portal, Admin Dashboard) (Employee Self Service)



Process used  
Microservices  
Architecture.  
as it's provides  
API's to  
operate diff.  
micro services  
within the  
rather than  
monolithic

Based  
on  
HR  
Domain

Application Layer

- Core HR Management  
- Payroll

Recruiting / Applicant Tracking  
- Performance & Feedback

Time &  
Attendance  
Tracking



Service Integration Layer

- API Gateway

- Authentication  
(OAuth 2, SAML)

- Message Queues / Event Bus



Data Access & Persistence Layer

- ORM / Database Access APIs.

- Caching (e.g. Redis)



Database Storage Layer

- RDBMS (MySQL)

- Document Storage (Amazon S3)

External Services

Payroll APIs  
(e.g. ADP)

Communication  
Slack, Team.

Calendar APIs  
Outlook

Identity  
Providers  
Azure AD,  
Google Workspace

## Ques 3 :

The 4+1 architectural view model includes the following views:

**Logical View:** This view shows the key abstractions of the system as objects or object classes. For a ticket machine, this would include classes representing concepts like:

Ticket (with attributes like type, price, origin, destination, validity)

Station (with attributes like name, location, available routes)

Route (connecting stations)

PaymentMethod (e.g., Cash, Card, Mobile Payment)

Transaction (recording details of a ticket purchase)

UserInterface (handling display and user input)

Printer (for printing tickets)

PaymentProcessor (interacting with payment gateways)

Timetable (containing schedule information) This view would typically be illustrated using UML Class Diagrams.

**Process View:** This view shows how the system is composed of interacting processes at run-time. For a ticket machine, this would illustrate processes like:

Starting the machine and initializing components.

User selecting a destination and ticket type.

System calculating the fare.

User choosing a payment method.

Processing the payment (interaction with external payment systems).

Printing the ticket.

Handling errors (e.g., payment failure, printer error). This view is often illustrated using UML Sequence Diagrams or Activity Diagrams.

**Development View:** This view shows how the software is decomposed for development. For a ticket machine, this would show the organization of code modules, packages, and components that developers would work on. Examples of components could be:

UI Component

Business Logic Component

Data Access Component

Printing Component

Payment Integration Component

Configuration Component This view is typically illustrated using UML Package Diagrams or Component Diagrams.

**Physical View:** This view shows the system hardware and how software components are distributed across the processors in the system. For a ticket machine, this would include hardware elements like:

The main processing unit.

Touch screen display.

Card reader.

Cash acceptor/dispenser.

Ticket printer.

Network interface (for communication with backend systems).

Sensors. The diagram would show which software components run on which hardware units. This view is illustrated using UML Deployment Diagrams.

The (+1) view is the Use Case View, which describes the system's functionality from the user's perspective. This would include use cases like "Purchase Single Ticket," "Purchase Return Ticket," "Check Train Schedule," "Top up Smart Card," etc.

View Type	Description
Logical View	Modules: UI, Payment, Ticket Generator, Backend Communication
Development View	Package structure, code modules (e.g., UI Layer, Controller, Service)
Process View	Concurrent components (UI thread, backend comms, card reader)
Physical View	Deployed on embedded Linux device, connects via Ethernet/WiFi to backend server