Applied Data Science

Project 2 Design Phase
Part -2

Group 4

Customer Journey Mapping :

- Customer journey mapping is a visual representation of the steps a customer takes to complete a specific action, such as signing up for a product trial or subscribing to a newsletter.
- the goal to improve the overall customer experience, customer journey maps help your company maintain a customer-centric mindset, identify any bottlenecks or siloes, and quickly spot what needs to be addressed. Businesses often have multiple customer journey maps, each reflecting a different area where the customer engages with your business or brand.

Benefits Of Customer Journey Mapping:

- Clarify channel performance
- Understand customer needs
- Improve decision-making
- Improve the customer experience

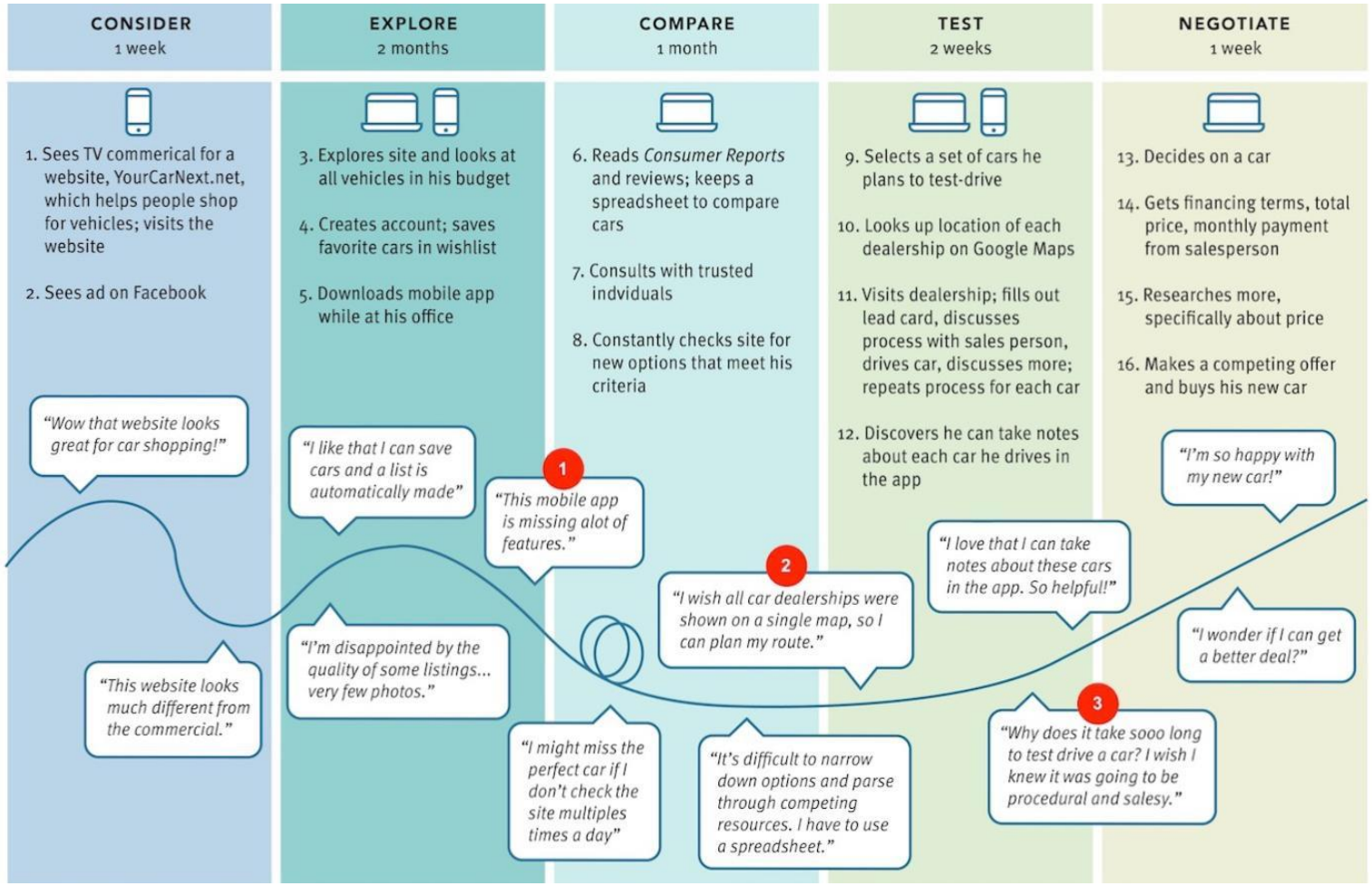
customer journey mapping tools:

- Stay proactive with a real-time view
- Create unique pipelines based upon customer behaviour
- Offer seamless, personalised experiences for your customers
- Improve the customer experience across all touch points
- Build customer journeys quickly with an easy-to-use interface

Build a Customer Journey Map:

- Set a clear objective for the map.
- Define your personas and highlight target customers.
- Define stages and identify goals for each.
- List out touch points.
- Gather data and customer feedback.
- Determine pain points and points of friction.
- Identify areas for improvement.

Customer Journey Map:



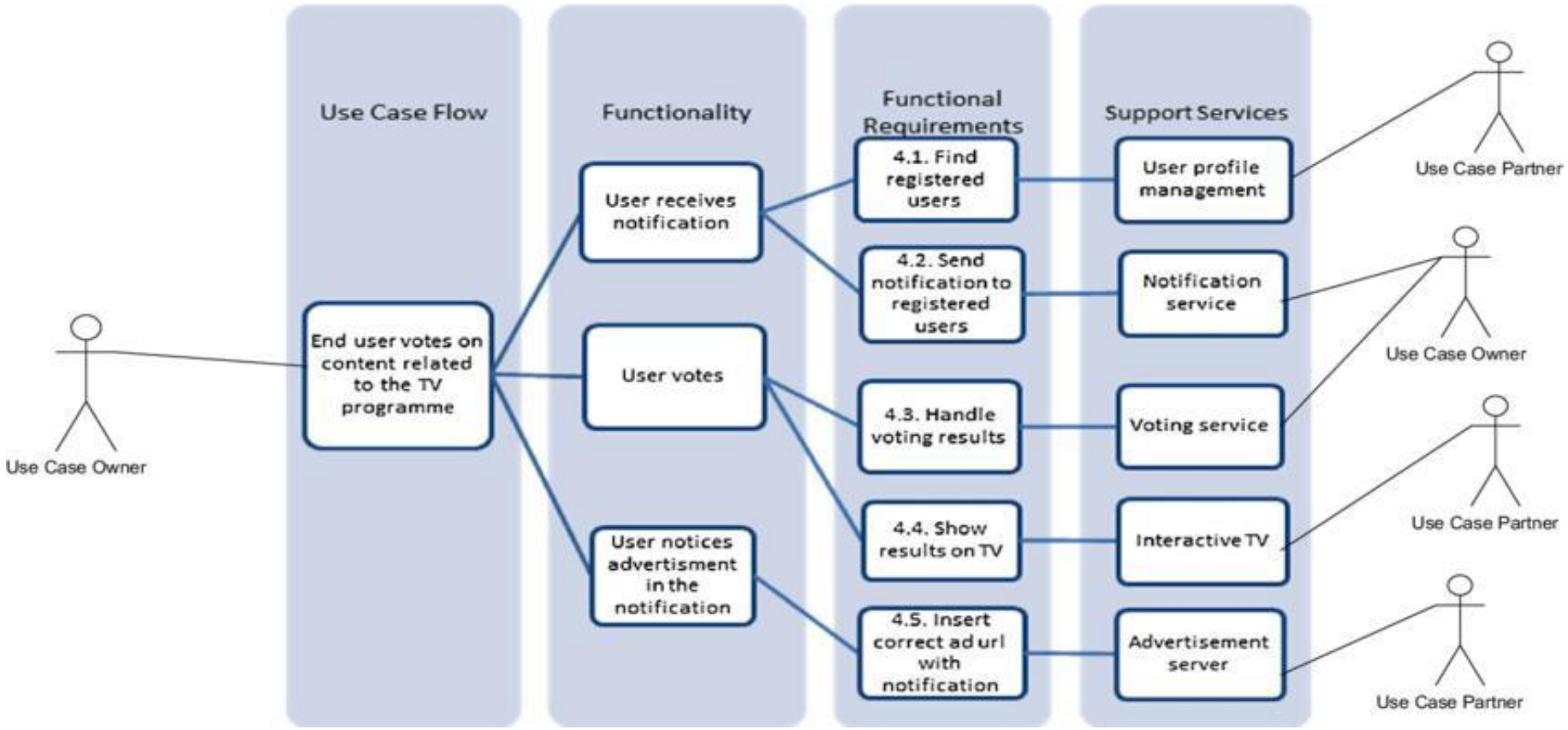
Functional Requirements:

- A functional requirement is a statement of how a system must behave. It defines what the system should do in order to meet the user's needs or expectations. Functional requirements can be thought of as features that the user detects.
- They are different from non-functional requirements, which define how the system should work internally (e.g., performance, security, etc.).

Types of Functional Requirements:

- Business Regulations
- Certification Requirements
- Reporting Requirements
- Administrative Functions
- Authorization Levels
- Audit Tracking
- External Interfaces
- Data Management
- Legal and Regulatory Requirements

Functional Requirement Analysis:



Input:

- *Create table for missing data analysis*
- def draw_missing_data_table(df):
- total = df.isnull().sum()
- sort_values(ascending=False)
- Percent=(df.isnull().sum()/df.isnull().count())
- sort_values(ascending=False) missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent’])
- return missing_data

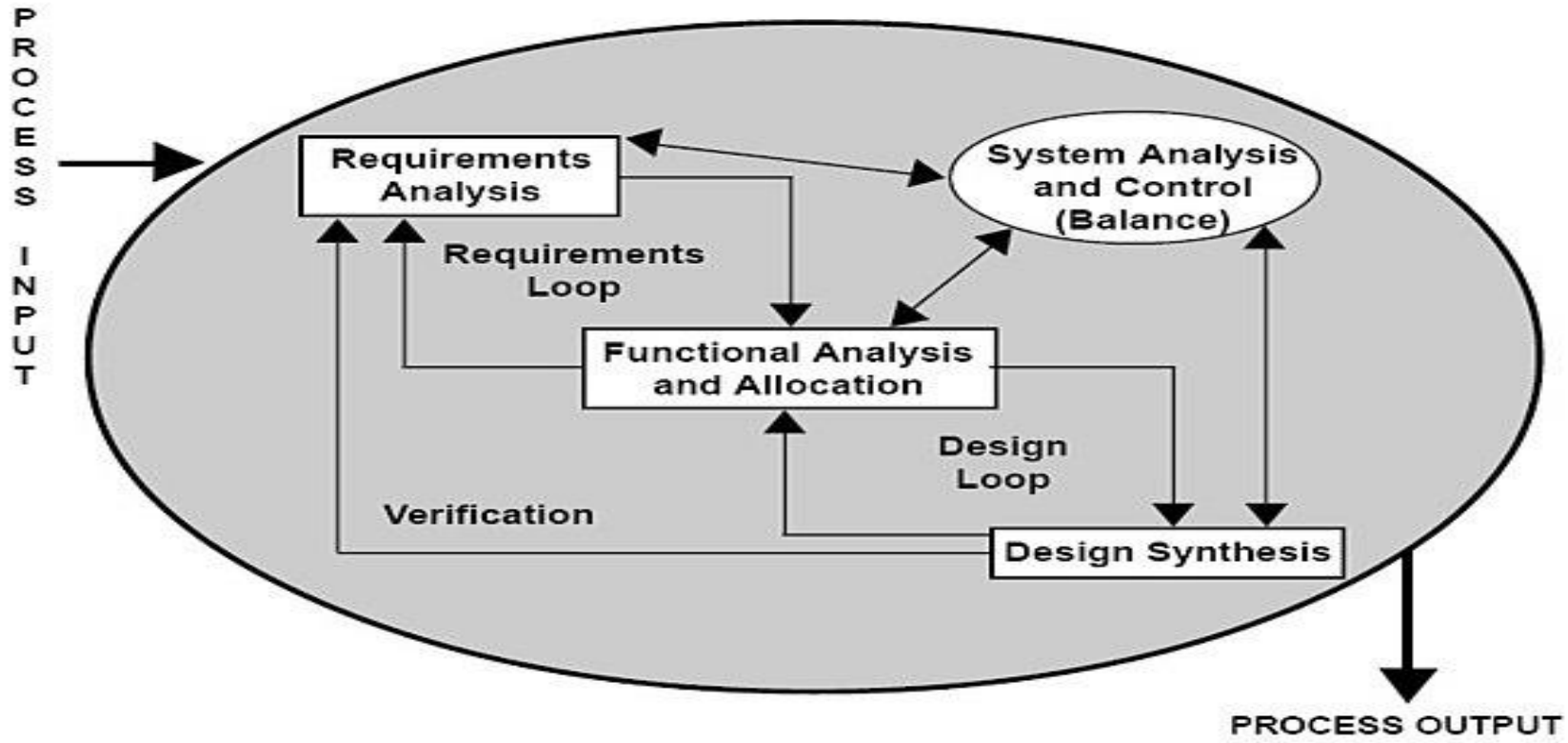
Output:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	

Requirement Analysis Operation:

- Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed.
- In software engineering, such requirements are often called functions specifications

Requirements Analysis:

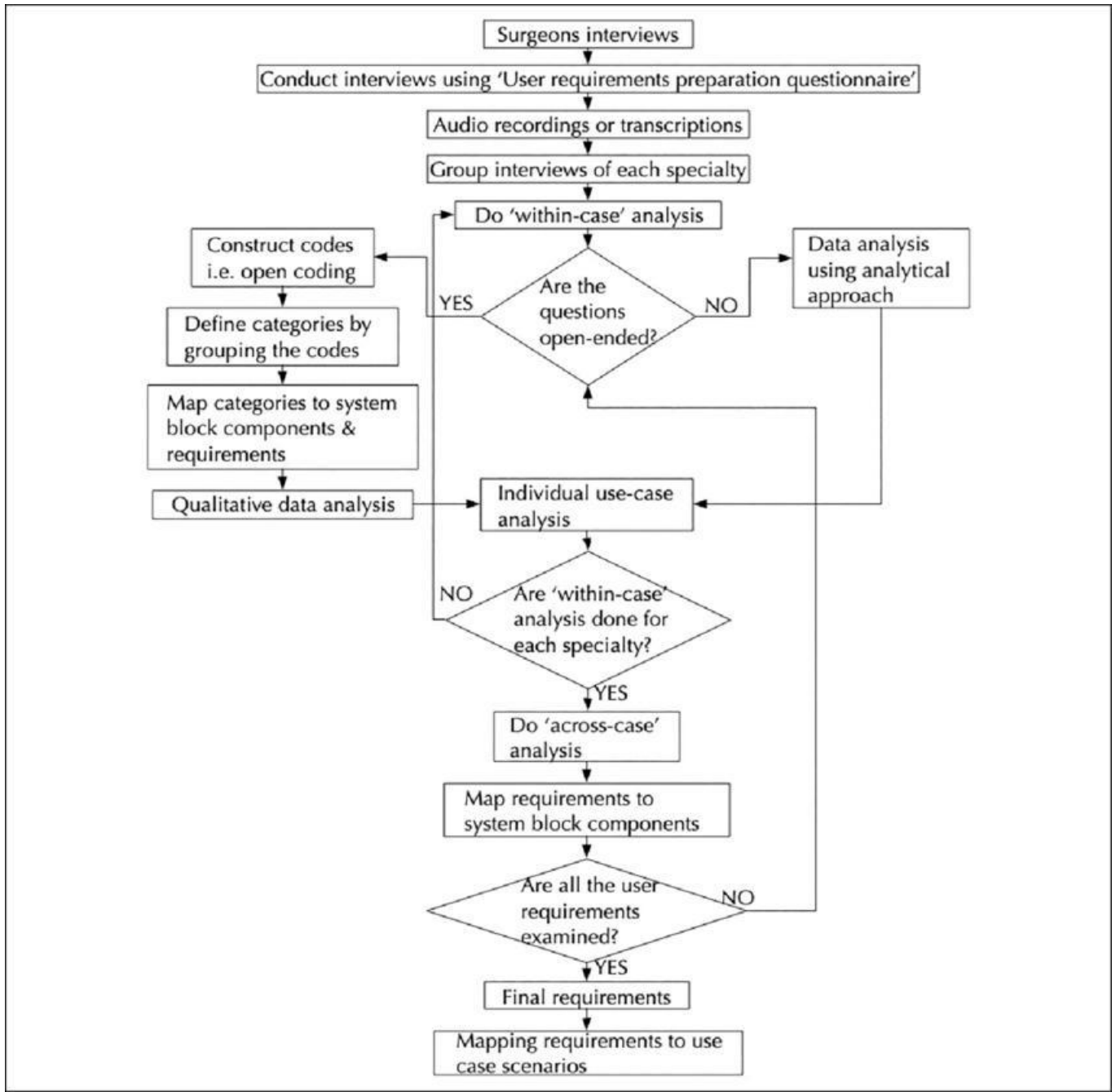


Requirement Analysis Technical:

- Business process modelling notation (BPMN)
- UML (Unified Modelling Language)
- Flowchart technique.
- Data flow diagram.
- Role Activity Diagrams (RAD)
- Gantt Charts.
- IDEF (Integrated Definition for Function Modelling)
- Gap Analysis.

Technical Analysis:

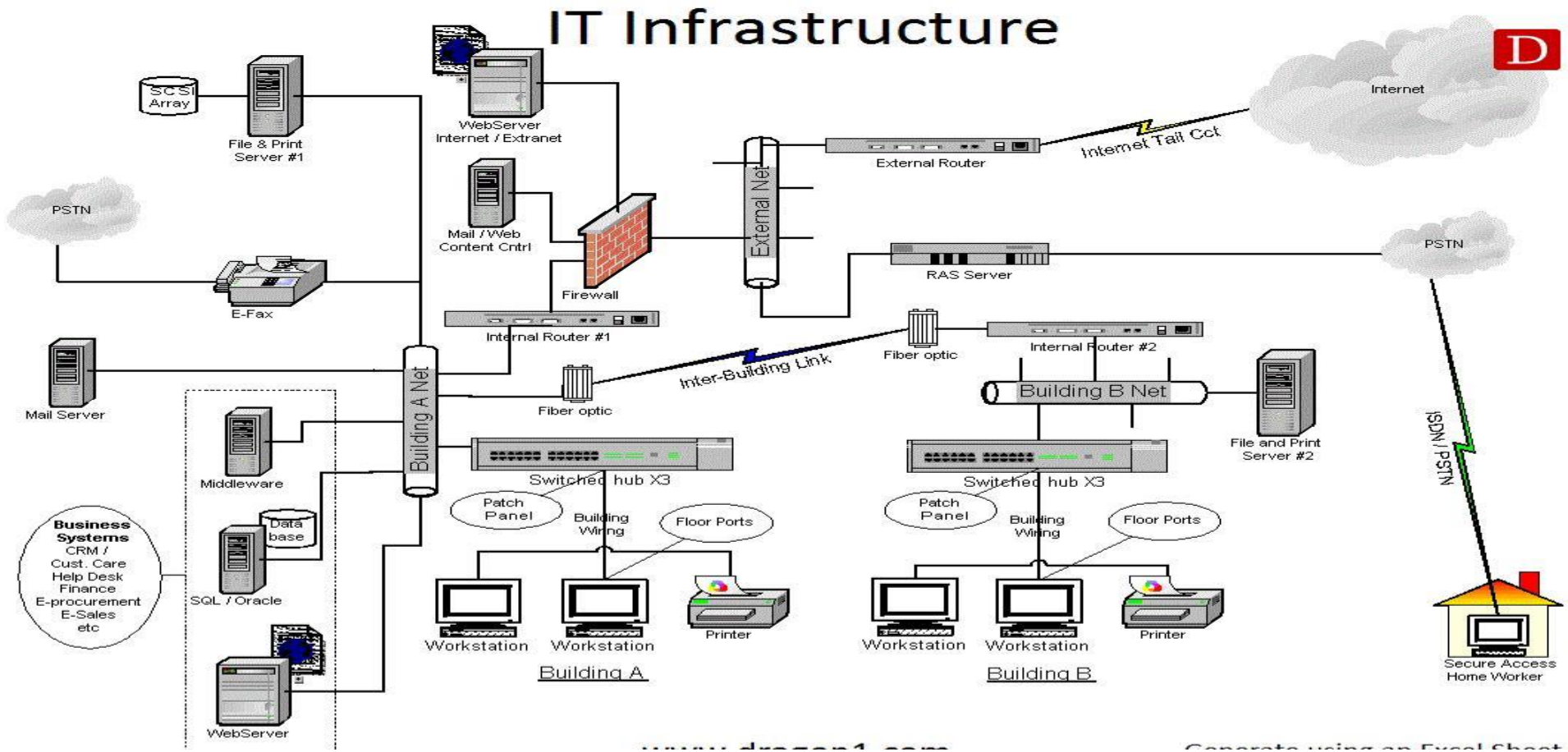
Flow Chart:


Surgeons interviews
Conduct interviews using 'User requirements preparation questionnaire'
Audio recordings or transcriptions
Group interviews of each specialty
Do 'within-case' analysis
Construct codes i.e. open coding
Define categories by grouping the codes
Map categories to system block components & requirements
Qualitative data analysis
Are the questions open-ended?
YES
NO
Data analysis using analytical approach
Individual use-case analysis
Are 'within-case' analysis done for each specialty?
NO
YES
Do 'across-case' analysis
Map requirements to system block components
Are all the user requirements examined?
NO
YES
Final requirements
Mapping requirements to use case scenarios

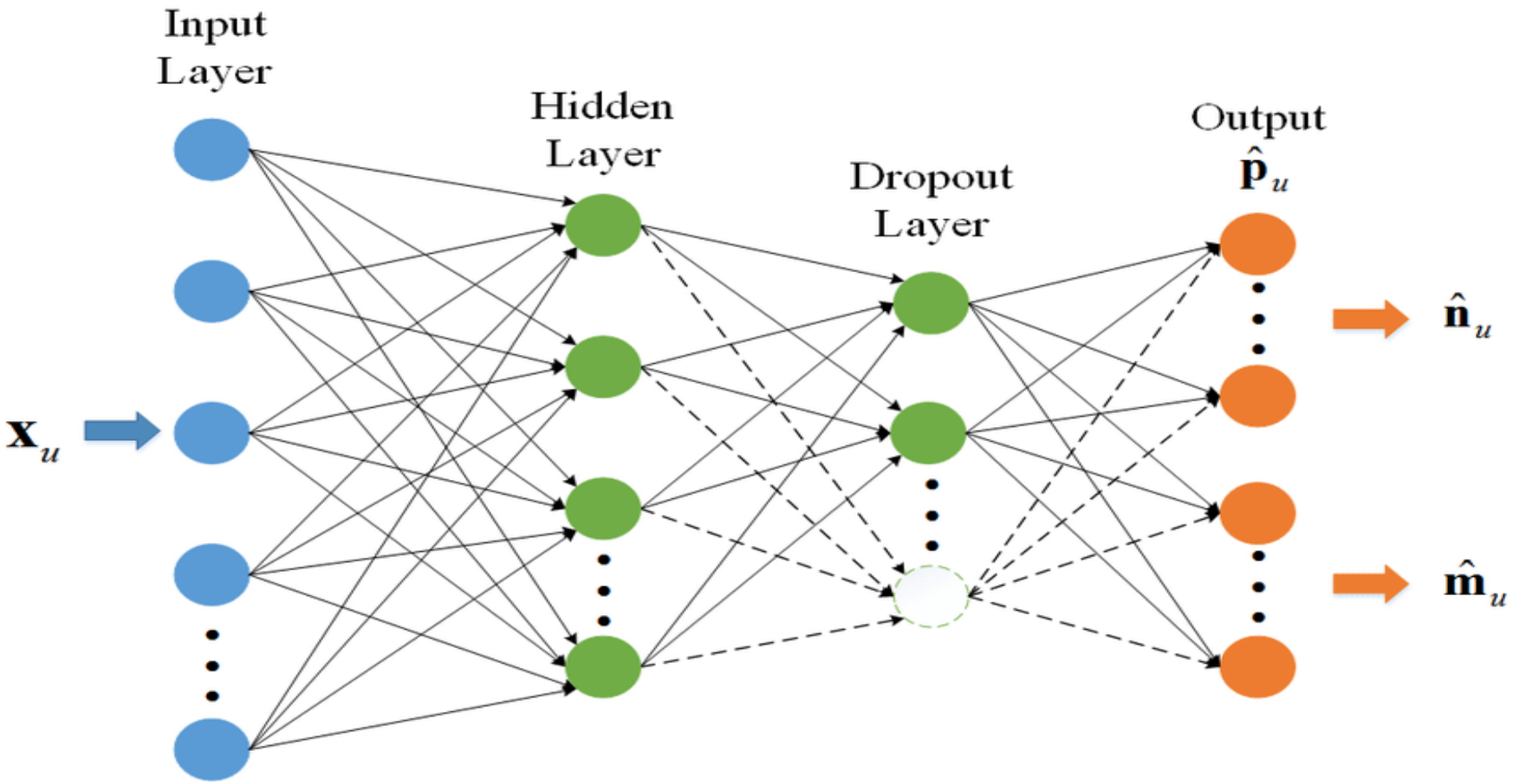
Technical Architecture:

- Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

Technical Architecture:



Architecture :



Input:

- log_dir="./logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
- tensorboard_callback=tf.keras.callbacks.TensorBoard(log_dir=log_dir) callback_list = [tensorboard_callback] *# start training* model.fit(train_generator, epochs=EPOCHS, steps_per_epoch=train_num
- // BATCH_SIZE
- validation_data=valid_generator, validation_steps=valid_num
- // BATCH_SIZE, callbacks=callback_list, verbose=0) model.summary()
- *# save the whole model*
- model.save(model_dir)

Output:

- Model: "alex_net"
- _____
- **Layer (type) Output Shape Param**
- **# =======**
- **==========**
- **conv2d (Conv2D) (None, 55, 55, 96) 34944**
- **_____**
- **max_pooling2d (MaxPooling2D) (None, 27, 27, 96) 0**
- **_____**
- **conv2d_1 (Conv2D) (None, 27, 27, 256) 614656**
- **_____**
- **max_pooling2d_1 (MaxPooling2 (None, 13, 13, 256) 0**
- **_____**
- **conv2d_2 (Conv2D) (None, 13, 13, 384) 885120**
- **_____**
- **conv2d_3 (Conv2D) (None, 13, 13, 384) 1327488**
- **_____**
- **conv2d_4 (Conv2D) (None, 13, 13, 256) 884992**
- **_____**
- **max_pooling2d_2 (MaxPooling2 (None, 6, 6, 256) 0**
- **_____ f**
- **latten (Flatten) (None, 9216) 0 _____**
- **dense (Dense) (None, 4096) 37752832 _____**
- **dense_1 (Dense) (None, 4096) 16781312 _____**
- **dense_2 (Dense) (None, 1000) 4097000 _____**
- **dense_3 (Dense) (None, 2) 2002 ==========**
- **Total params: 62,380,346 Trainable params: 62,380,346 Non-trainable par**ams: 0 _____

Open Source:

- Open source software is software with source code that anyone can inspect, modify, and enhance
- Source code" is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application"—works. Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly

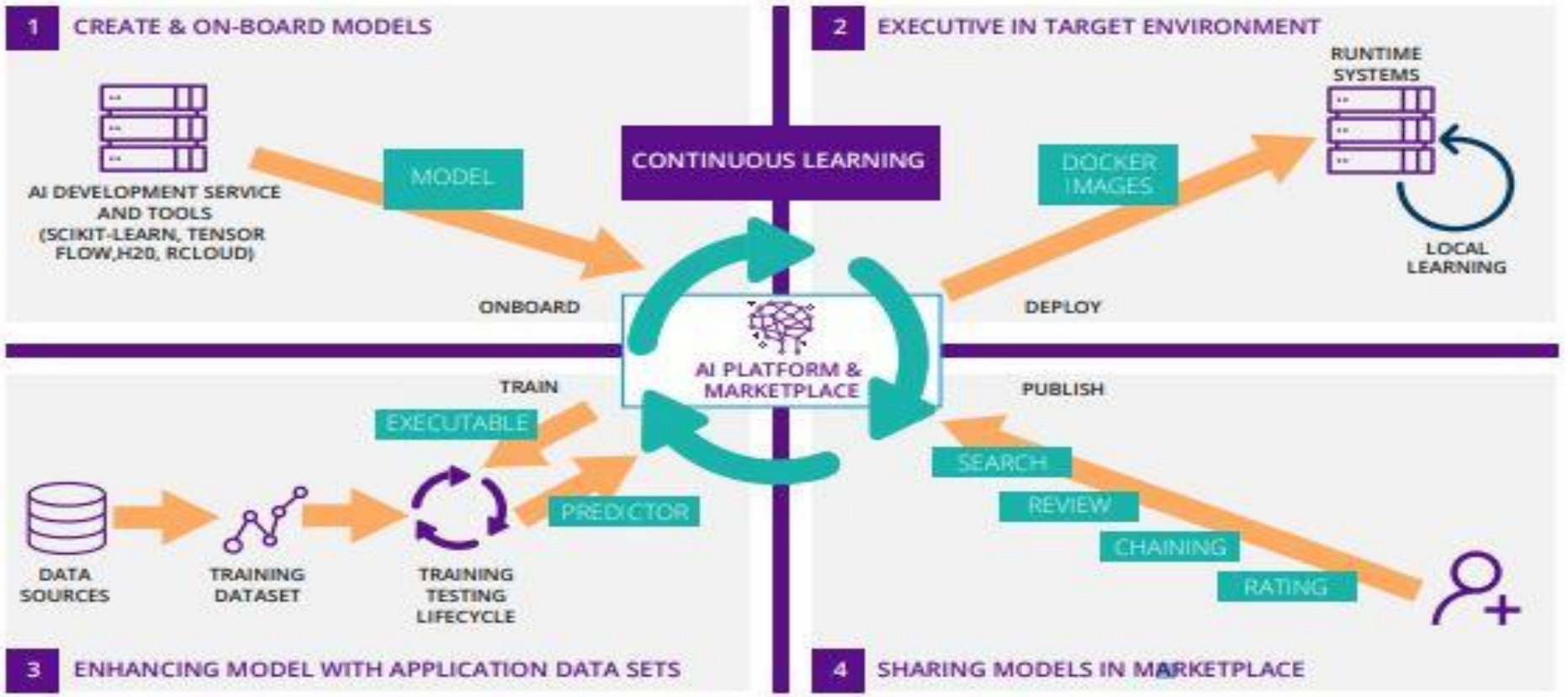
Open Source Frame Work:

- Open source is source code that is made freely available for possible modification and redistribution. Products include permission to use the source code, design documents, or content of the product. The open-source model is a decentralized software development model that encourages open collaboration

Open Source Examples:

- GNU/Linux.
- Mozilla Firefox.
- VLC media player.
- Sugar CRM.
- GIMP.
- VNC.
- Apache web server.
- Libreoffice

Diagram for Open source Framework:



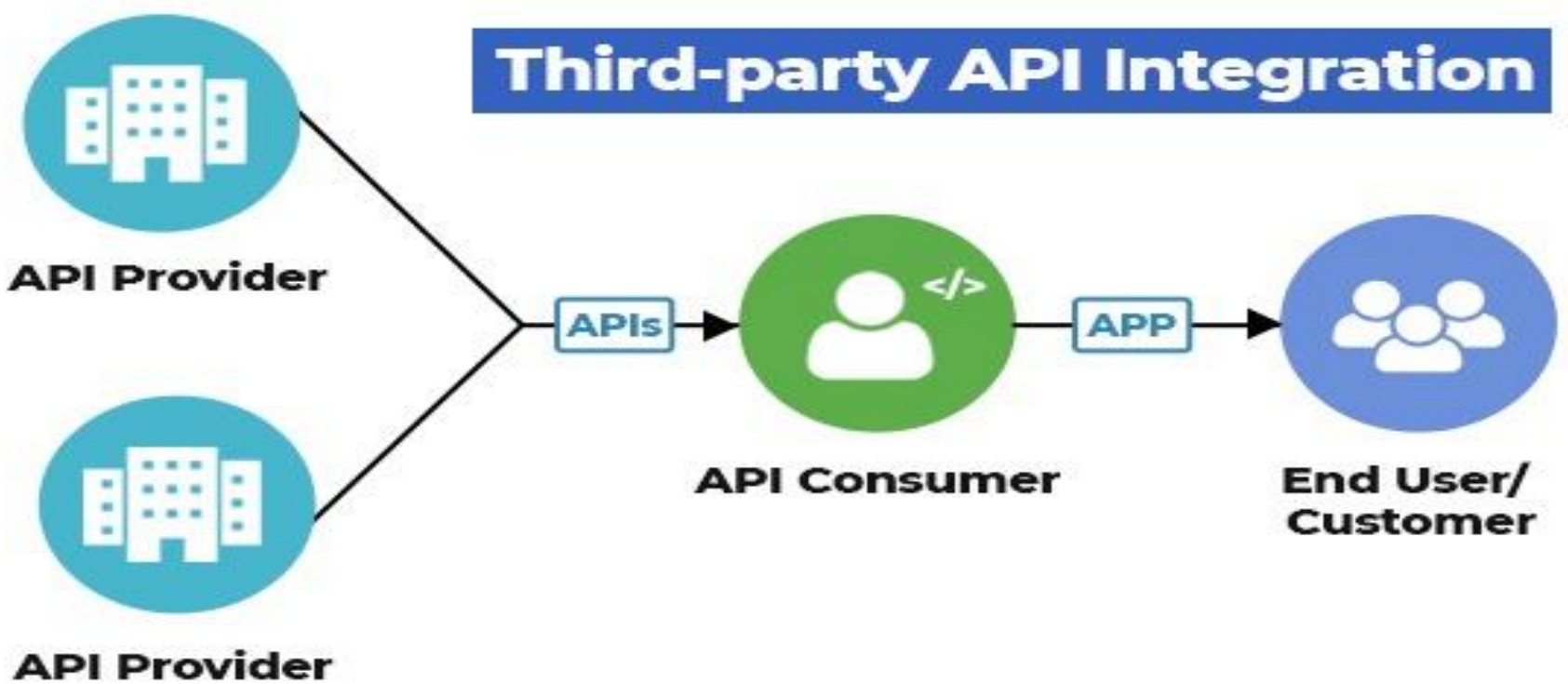
Third-Party API’S:

- Third party APIs are APIs provided by third parties — generally companies such as Facebook, Twitter, or Google — to allow you to access their functionality via JavaScript and use it on your site. One of the most obvious examples is using mapping APIs to display custom maps on your pages.

API integration to your mobile app:

- Hire an API integration developer. ...
- Create the project within the API provider system. ...
- Receive API key and authorization token. ...
- Integrate the API framework for the app. ...
- Use API request instances and methods.

Api integration:



Cloud Deployment:

- Cloud deployment is the process of deploying an application through one or more hosting models—software as a service (SaaS), platform as a service (PaaS) and/or infrastructure as a service (IaaS)—that leverage the cloud. This includes architecting, planning, implementing and operating workloads on cloud.

Types Of Cloud Deployment:

- SaaS,
- IaaS,
- PaaS
- Cloud Deployments Model:

public

private

virtual private (VPC),

hybrid

community cloud.

Cloud Deployment Model: