Chapter 1: Introduction

1.1 Brief overview of the work

VuMaxDR is desktop application and proposed task is to re-design and develop one of the application modules (Business Intelligence/KPI module) to web based solution using React with Typescript framework.

The BI module will be graphical frontend to the VuMaxDR System for the analysis of multi-well performance KPIs. The system takes advantage of the raw and derived data stored in the VuMax database to produce graphical and numerical summaries of the operation. The BI module uses several sources for the KPIs:

- Real-time data stored for the wells.
- Data Aggregates derived from the real-time data.
- Automatically tagged activities: Trips, CUCs.
- User Tagged Activities and KPIs for: Phases, Steps, and Emphasis.
- Drilling and Tripping connection times.

It allows the users to interactively define the operational windows that will be used in the summaries

1.2 Objective

Develop a web-based system that provides insight to the data stored in the VuMaxDR database. The system should provide advanced analytics and visualizations capabilities along with the ability to design the templates in a very user-friendly manner. The system should also provide the ability to generate report outputs in various formats like PDF and PowerPoint.

1.3 Scope

- System Design and Architecture:
  - Develop a comprehensive system architecture that outlines the components, modules, and data flow.
- Design interfaces for
  - ➔ Report templates
  - ➔ Well list
  - ➔ Execute templates against set of wells and generate output.
  - ➔ Drill down
- Analytics
  - Implement custom algorithms to perform advanced analytics on the data and produce results.
  - Implement standard functions
- User Management and Access Control:

- Design user authentication and authorization mechanisms to ensure secure access to data and features.
- Implement role-based access control (RBAC) to manage user permissions and privileges.

- Mobile Compatibility:
    - Optimize the web-based system for mobile devices to provide on-the-go access to critical business insights.

1.4 Project Modules

- User SignUp and Login : This module provides sign in and login facilities of google authentication.
- Search Blogs: Through this module, user can search the blog of their interest.
- User Profile: Using this module user profile will show and keep a track of user information.
- Web Scrappers for Blog Websites: Web scrappers will scrape the blogs from blogging website and will find out which are the topics that are covered in it using Natural Language Processing. This will enable us to recommend track the user interest and recommend the blogs according to it.
- Blog API: The blog API will provide us the latest blogs that are obtained from web scrapping different blogs.
- Recommendation System: The Recommendation System will track the user activity and according to that it will recommend the blogs to them.
- Likes: In this module , User can show the blogs which is liked by them
- Favourites: In this module, User can show their favorite blogs which is added by user.

1.5 Project Hardware/Software Requirements

1.5.1 Software:

1.5.1.1 Frontend:

- React js: for creating and structuring content for the Website.
- Typescript: Extend the JavaScript and improve the developer experience
- Material-UI: includes a comprehensive collection of prebuilt components that
- are ready for use in production right out of the box

1.5.1.2 Backend:

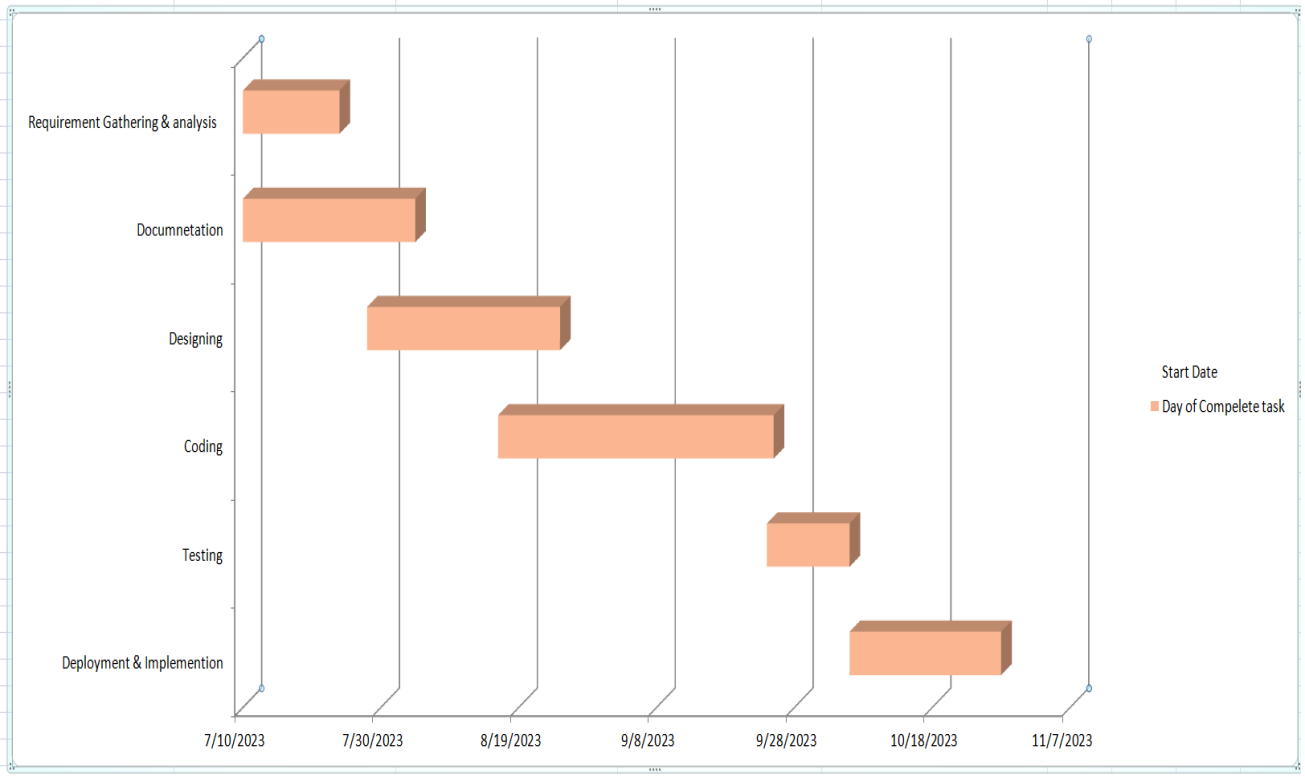
- C#: C-Sharp is one of the most widely used languages for creating system backend.

1.5.1.3 Database:

- MySQL: To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server**.**

Chapter 2: System Analysis & Design

2.1 Project Timeline Chart:



2.2 Project SRS:

2.2.1 Use Case Diagram:

Use case diagrams are a common way to communicate the major functions of a software system. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Use cases are nothing but the system functionalities written in an organized manner. Now another thing which is relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

So, in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

Symbols used in Use Case Diagram:


Use Case


Include

Association

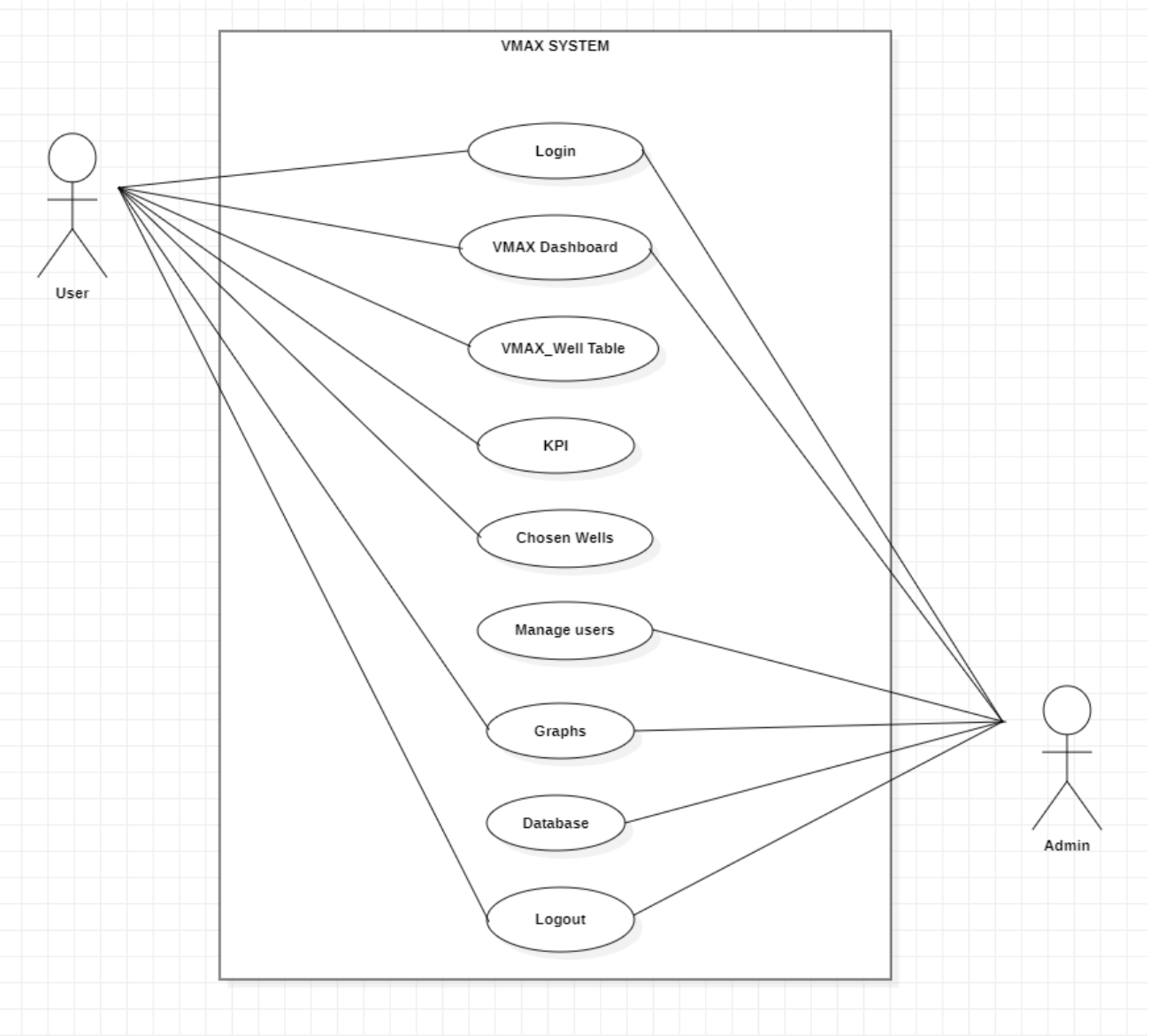

Extend


Actor


Dependency


System


Generalization


VMAX SYSTEM
Login
VMAX Dashboard
VMAX_Well Table
KPI
Chosen Wells
Manage users
Graphs
Database
Logout
User
Admin

2.2.2 Class Diagram:

The class diagram is the main building block of object-oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

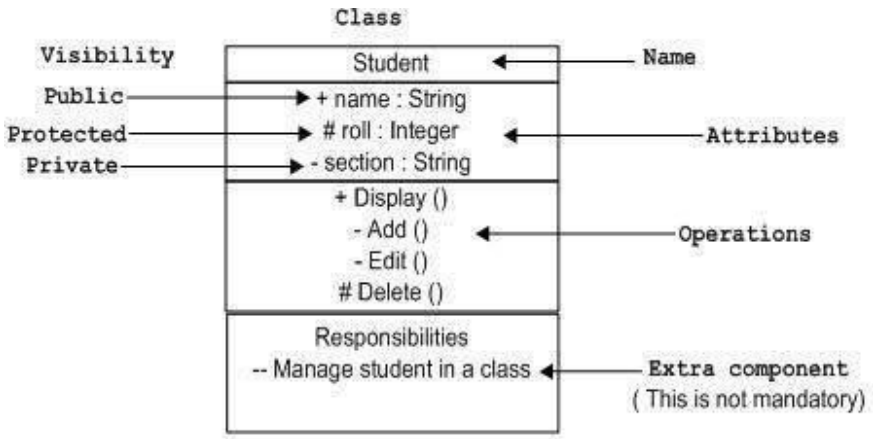
In the diagram, classes are represented with boxes that contain three compartments:

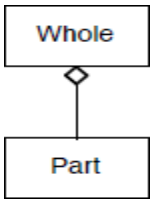
- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

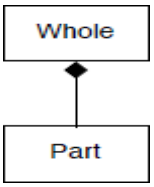
In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a number of subclasses.

Symbols used in Class Diagram:

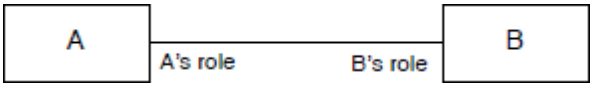
Class

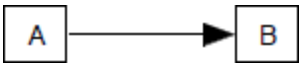






Association

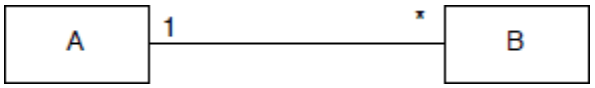


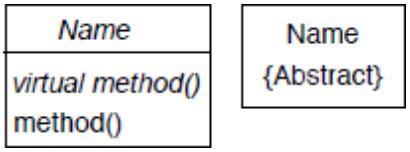


Multiplicity in Aggregation,

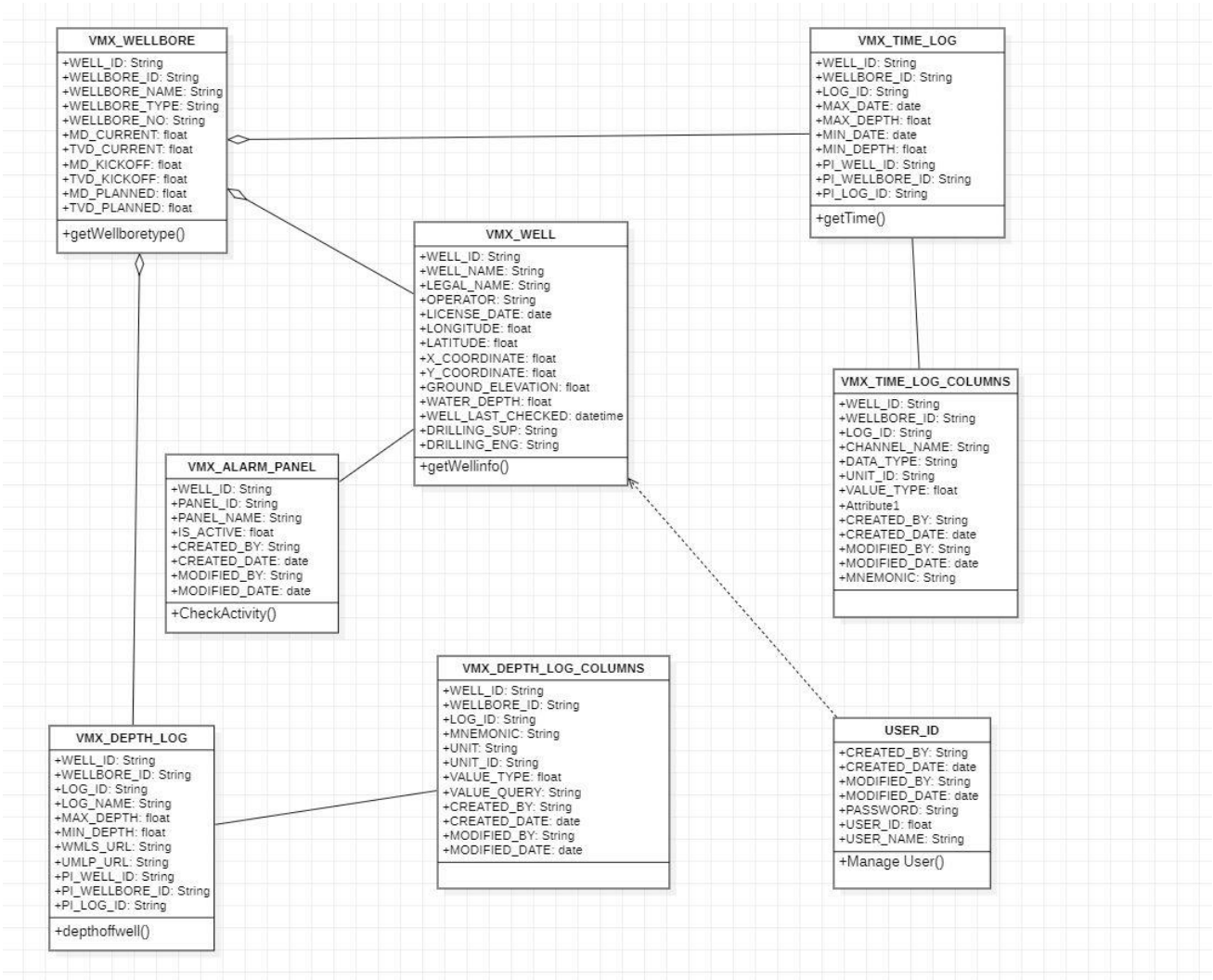
Abstract Class

Composition, or Association





Class Diagram (Blog Recommender System):



2.2.3 E-R Diagram:

E-R Model Full Form: Entity-Relationship Model.

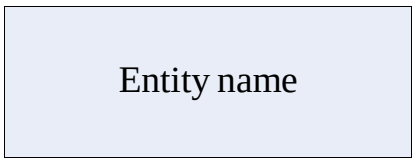
Entity-Relationship model is used to represent a logical design of a database to be created. In ER model, real world objects (or concepts) are abstracted as entities, and different possible associations among them are modeled as relationships.

We represent the attributes, entities and relation using the ER diagram. Using this ER diagram, table structures are created, along with required constraints.

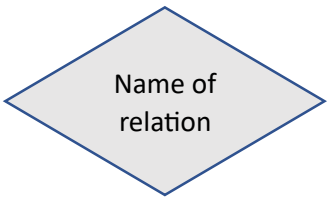
Finally, these tables are normalized in order to remove redundancy and maintain data integrity. Thus, to have data stored efficiently, the ER diagram is to be drawn as much detailed and accurate as possible.

Symbols used in ER diagram:

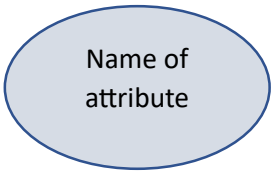
Entity



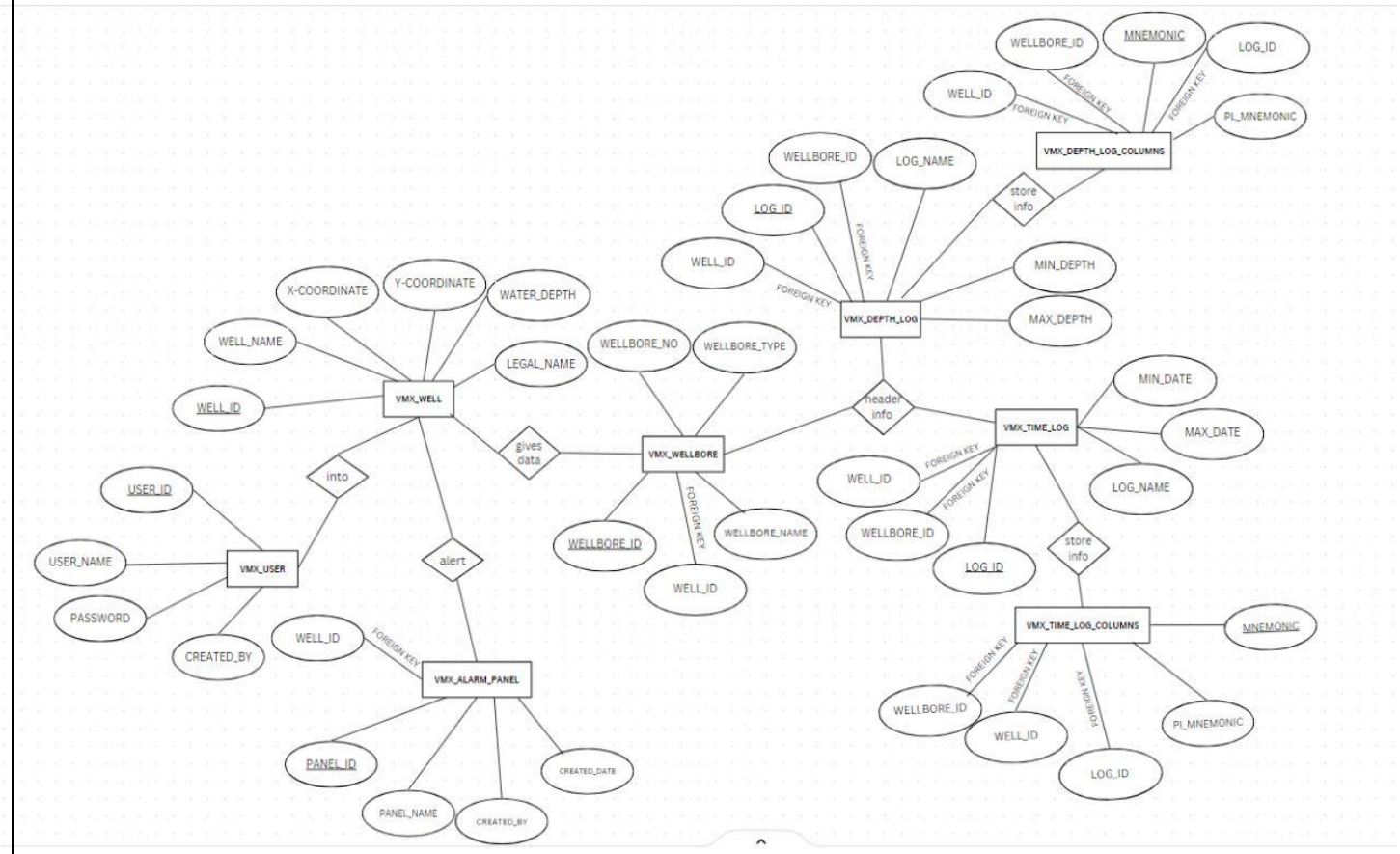
Relation



Attributes



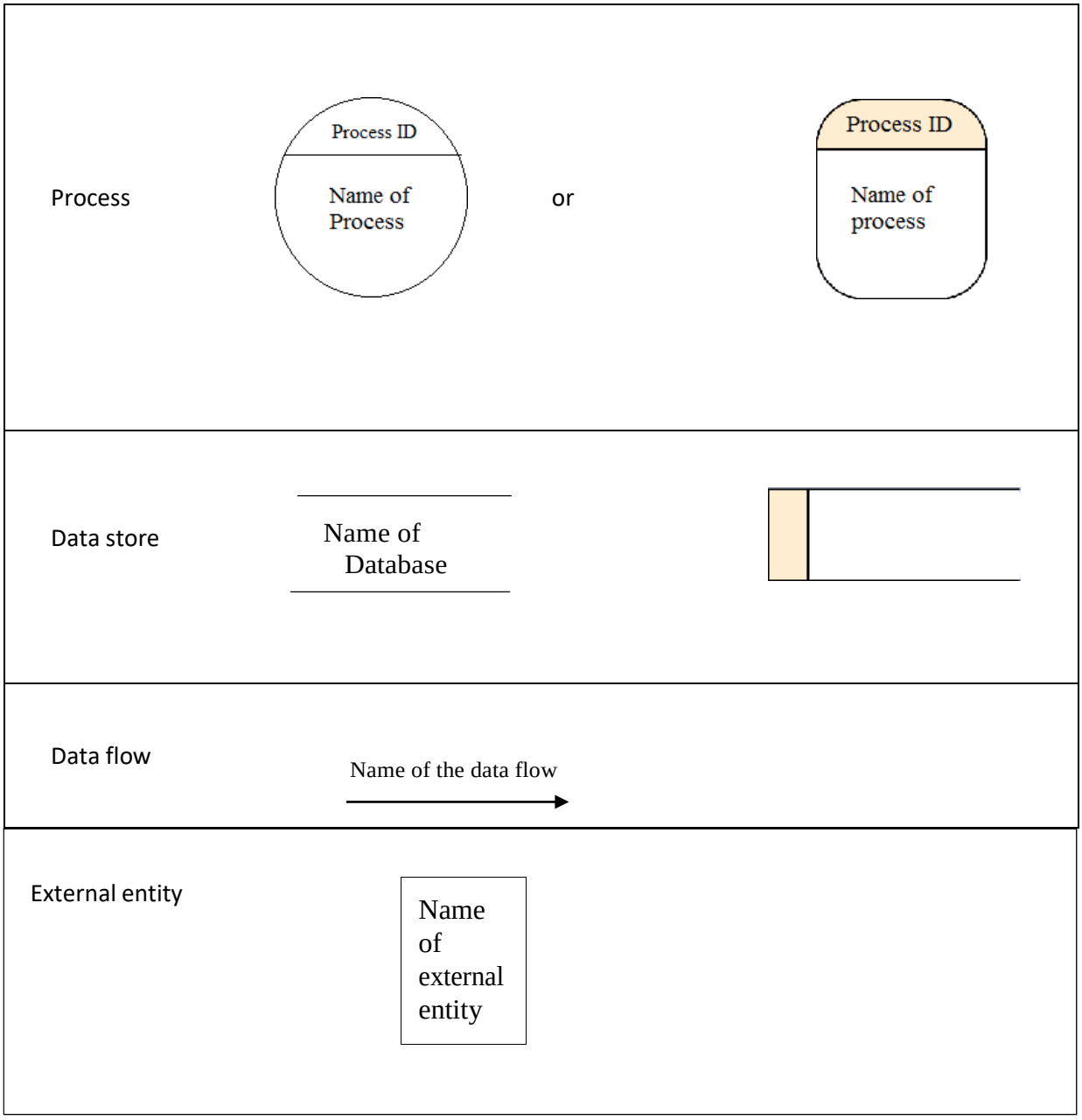
Entity–Relationship Diagram (Blog Recommendation System):



2.2.4 Data Flow Diagram:

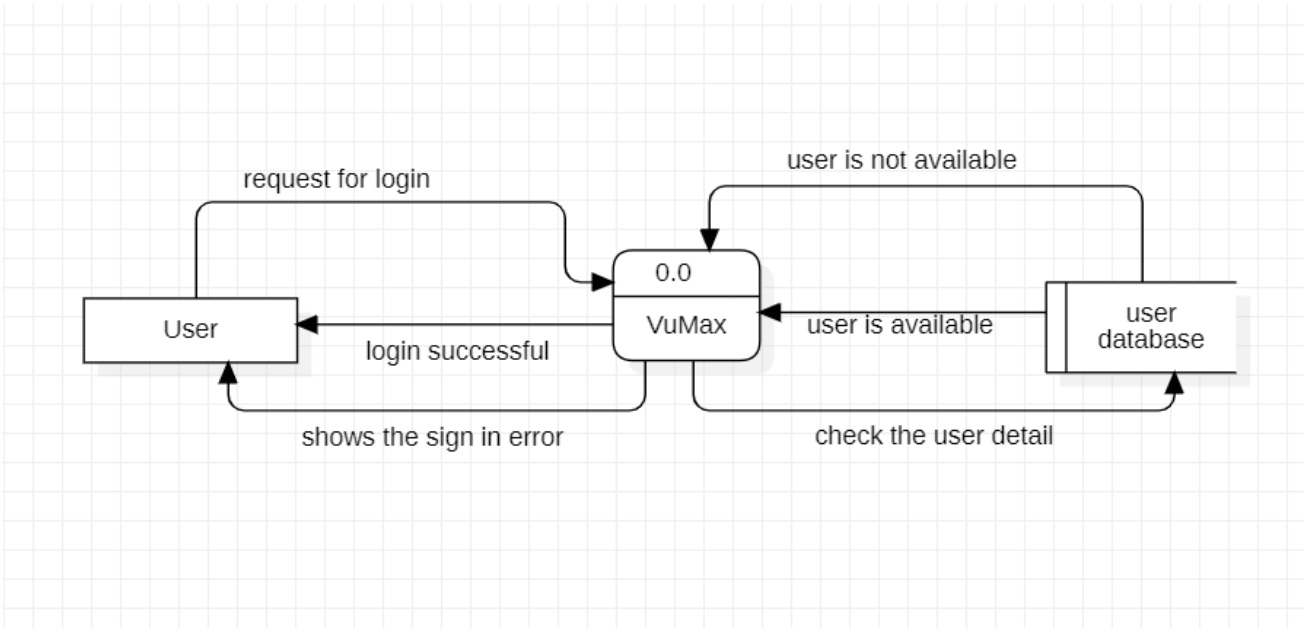
DFD provides the functional overview of a system. The graphical representation easily overcomes any gap between 'user and system analyst' and 'analyst and system designer' in understanding a system. Starting from an overview of the system it explores detailed design of a system through a hierarchy. DFD shows the external entities from which data flows into the process and also the other flows of data within a system. It also includes the transformations of data flow by the process and the data stores to read or write a data.

Symbols used in Data Flow diagram:

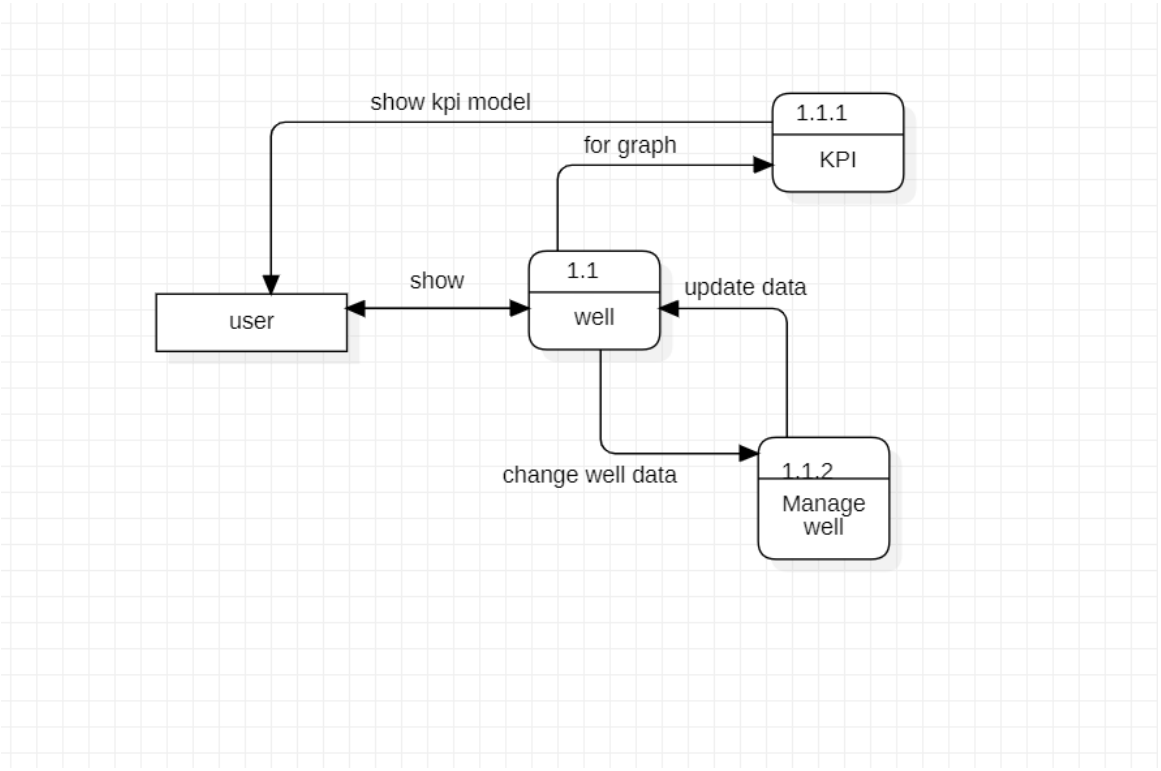


Data Flow Diagram of the System:

Level 0:



Level 1:



Level 2:

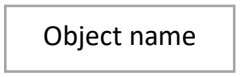
2.2.5 Sequence Diagram:

Sequence diagram represents the behavioral aspects of a system. Sequence diagram shows the interactions between the objects by means of passing messages from one object to another with respect to time in a system.

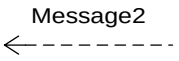
Sequence diagram contains the objects of a system and their life-line bar and the messages passing between them. Objects appear at the top portion of sequence diagram. Object is shown in a rectangle box. Name of object precedes a colon ':' and the class name, from which the object is instantiated. The whole string is underlined and appears in a rectangle box. A down- ward vertical line from object-box is shown as the life-line of the object. A rectangle bar on life-line indicates that it is active at that point of time. Messages are shown as an arrow from the life-line of sender object to the life-line of receiver object and labelled with the message name.

Symbols used in Sequence diagram:

Object

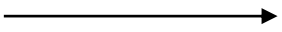


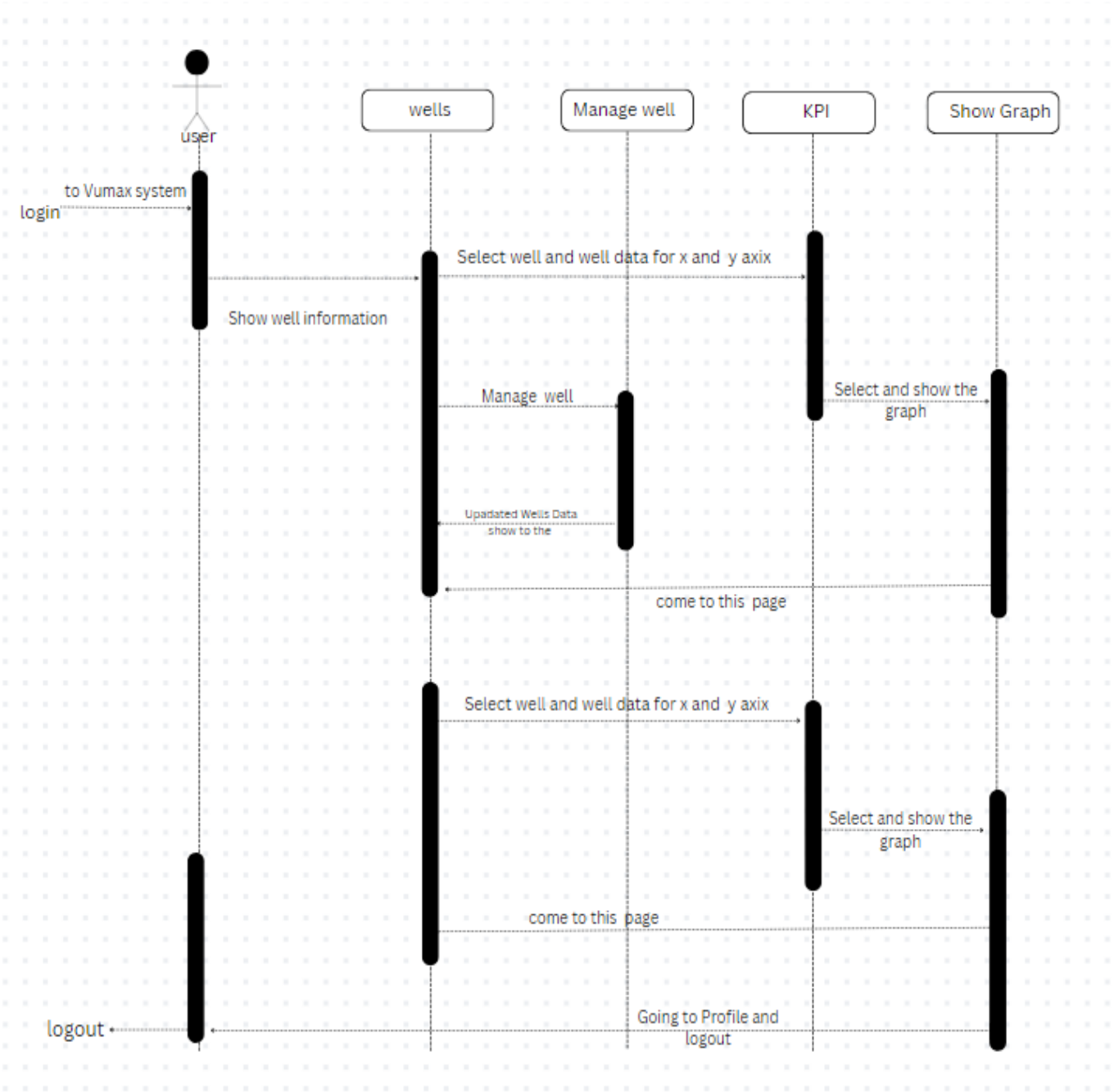
Return message



Life line bar

Call message




user
wells
Manage well
KPI
Show Graph
to Vumax system
login
Select well and well data for x and y axix
Show well information
Manage well
Select and show the
graph
Upadated Wells Data
show to the
come to this page
Select well and well data for x and y axix
Select and show the
graph
come to this page
logout
Going to Profile and
logout

2.2.6 State Diagram:

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system. A State diagram describes a state machine. Now to clarify it, state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. As State diagram defines states it is used to model lifetime of an object.

State diagram is one of the UML diagrams used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. So State diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

State diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State diagram is to model life time of an object from creation to termination. State diagrams are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.

Following are the main purposes of using State diagrams:

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model states of an object.

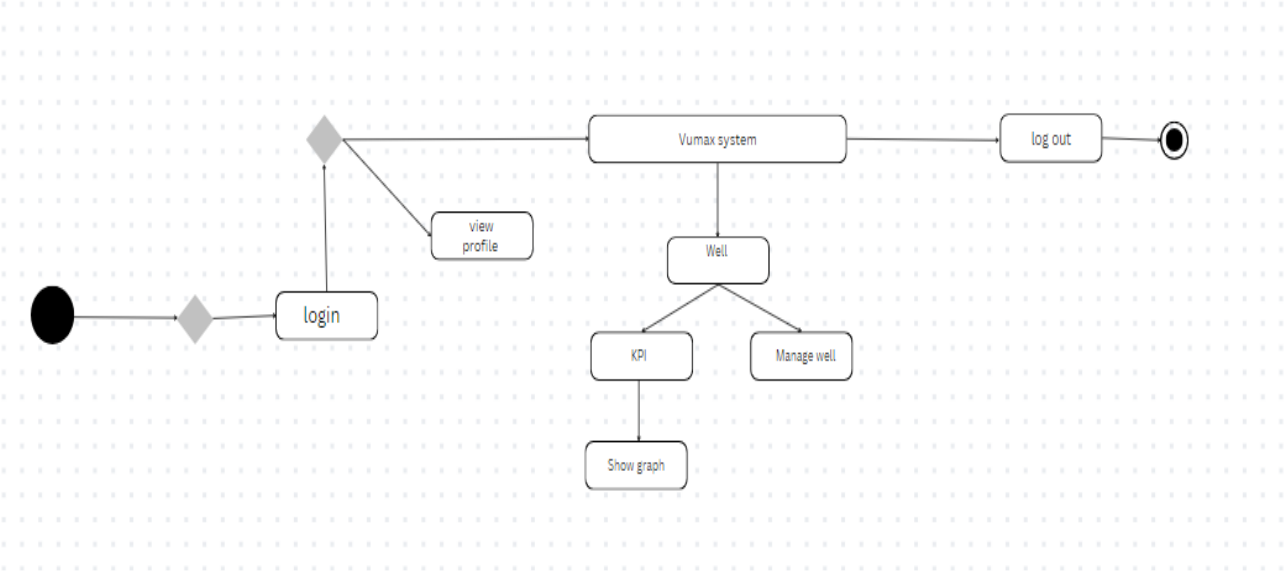
Symbols used in State diagram:

Start symbol

State symbol

End symbol

State Diagram :



2.2.7 Activity Diagram:

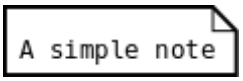
An activity denotes a particular action taken in the logical flow of control. This could simply be invocation of a mathematical function, alter an object's properties and so on. An activity is represented with a rounded rectangle, as shown in figure. A label inside the rectangle identifies the corresponding activity.

There are two special types of activity nodes: initial and final. They are represented with a filled circle, and a filled in circle with a border respectively. Initial node represents the starting point of a flow in an activity diagram. There could be multiple initial nodes, which mean that invoking that particular activity diagram would initiate multiple flows. A final node represents the end point of all activities. Like an initial node, there could be multiple final nodes. Any transition reaching a final node would stop all activities. A flow is represented with a directed arrow. A decision node, represented with a diamond, is a point where a single flow enters and two or more flows leave. This is represented with a diamond shape, with two or more flows entering, and a single flow leaving out. Fork is a point where parallel activities begin. A join is depicted with a black bar, with multiple input flows, but a single output flow. Physically it represents the synchronization of all concurrent activities.

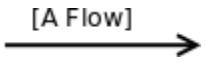
Symbols used in Activity diagram:

Activity

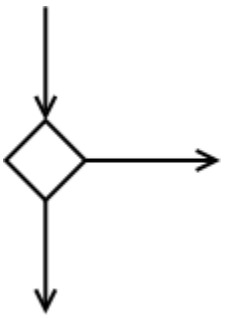
Note



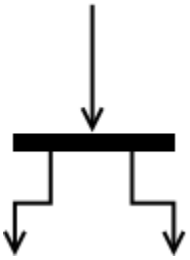
Flow



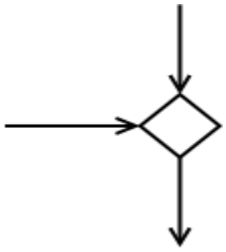
Decision



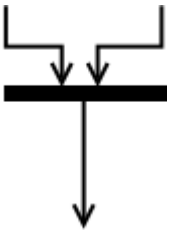
Fork

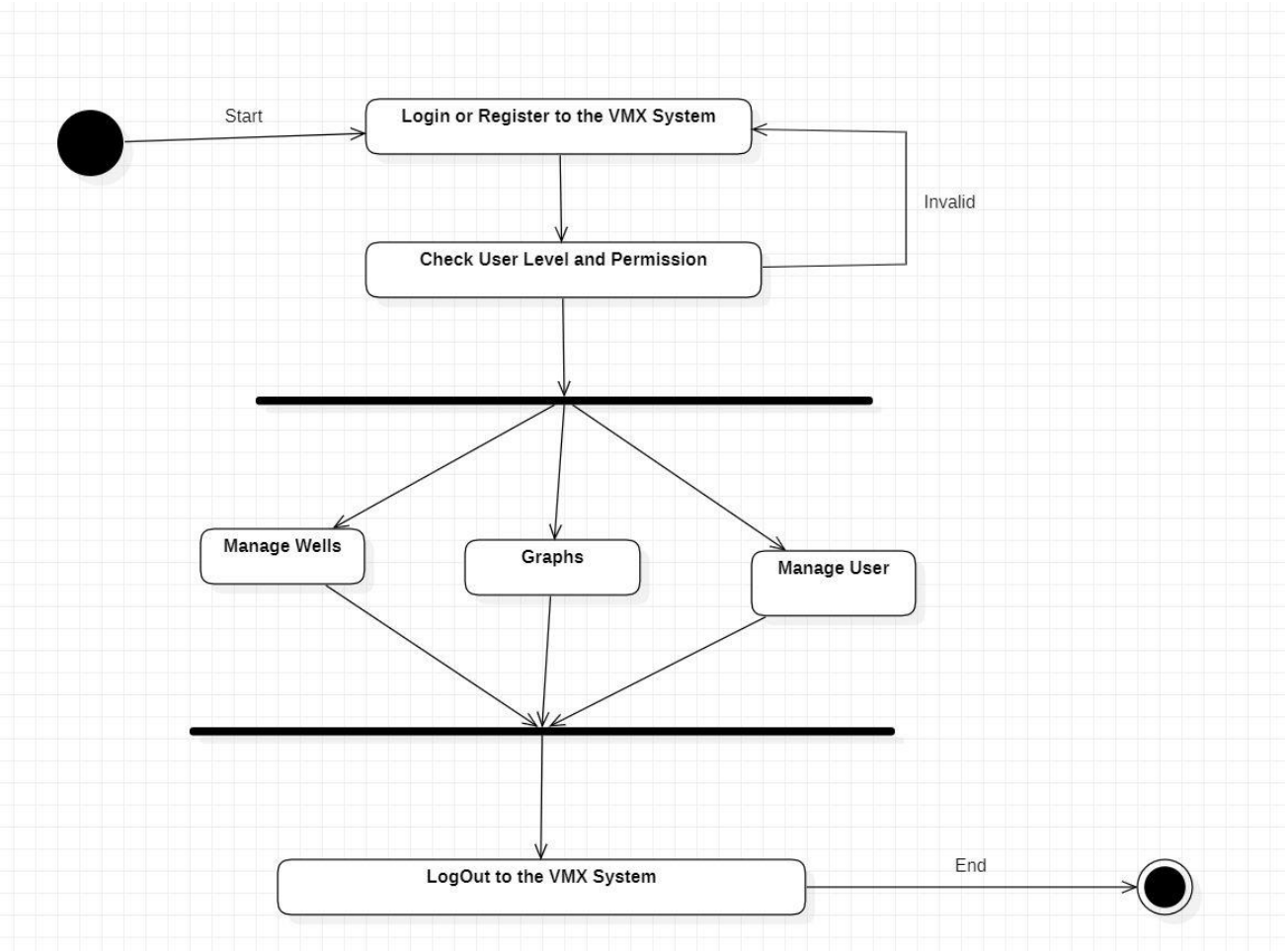


Merge



Join




Start
Login or Register to the VMX System
Invalid
Check User Level and Permission
Manage Wells
Graphs
Manage User
LogOut to the VMX System
End

2.3 Data Dictionary:

2.6.1 Table Name: VMX_WELL
Description: This table stores Well Header Information.
Primary Key: WELL ID

Sr. No.	Name	Datatype	Constraint	Description
1	WELL_ID	varchar(100)	Primary Key	Well ID
2	UWI	varchar(100)	Not Null	Unique Well ID
3	WELL_NAME	varchar(500)	Not Null	Well Name
4	LEGAL_NAME	varchar(100)	Not Null	Well Legal Name
5	BLOCK	varchar(500)	Not Null	Block
6	FIELD	varchar(500)	Not Null	Field
7	COUNTY	varchar(500)	Not Null	County
8	DISTRICT	varchar(500)	Not Null	District
9	REGION	varchar(500)	Not Null	Region
10	STATE	varchar(500)	Not Null	State
11	COUNTRY	varchar(500)	Not Null	Country
12	OPERATOR	varchar(500)	Not Null	Well Operator Name
13	OPERATOR_DIV	varchar(100)	Not Null	Well Operator Division
14	LICENSE_NO	varchar(100)	Not Null	Well License No.
15	LICENSE_DATE	date	Not Null	Well License Date
16	PURPOSE	varchar(500)	Not Null	Purpose Of Well
17	STATUS	varchar(500)	Not Null	Well Status: Active / Inactive
18	SPUD_DATE	date	Not Null	Spud Date
19	PA_DATE	date	Not Null	PA Date
20	TIME_ZONE	varchar(50)	Not Null	Well Time zone
21	LONGITUDE	numeric(16, 5)	Not Null	Longitude
22	LATITUDE	numeric(16, 5)	Not Null	Latitude
23	X_COORDINATE	numeric(16, 5)	Not Null	X Co-Ordinate
24	Y_COORDINATE	numeric(16, 5)	Not Null	y Co-Ordinate
25	PERM_DATUM	varchar(100)	Not Null	Permanent Datum
26	WELL_HEAD_ELEVATION	numeric(16, 5)	Not Null	Well Head Elevation
27	GROUND_ELEVATION	numeric(16, 5)	Not Null	Ground Elevation
28	WATER_DEPTH	numeric(16, 5)	Not Null	Water Depth
29	WMLS_URL	varchar(500)	Not Null	Well WMLS Server URL
30	WMLP_URL	varchar(500)	Not Null	Well WMLP Server URL
31	ALARM_HISTORY_TABLE	varchar(100)	Not Null	Well Alarm History Table
32	CREATED_BY	varchar(50)	Not Null	Name of the user who created the record
33	CREATED_DATE	datetime	Not Null	Created Date

34	MODIFIED_BY	varchar(50)	Not Null	Name of the user who last modified the record
35	MODIFIED_DATE	datetime	Not Null	Modification Date
36	RIG_NAME	varchar(500)	Not Null	Rig Name
37	WELL_LAST_CHECKED	datetime	Not Null	Well Last Checked Date Time
38	WELL_FLAGGED	decimal(1, 0)	Not Null	Well Flagged : Yes / No
39	DATA_SOURCE	varchar(500)	Not Null	Data Source
40	ALARM_PROFILE_ID	varchar(100)	Not Null	Well Alarm Profile ID
41	DATE_FORMAT	varchar(5)	Not Null	Date Format For Well : UTC, Local...
42	EDR_PROVIDER	varchar(255)	Not Null	EDR Provider Name
43	DRILLING_SUP	varchar(500)	Not Null	Drilling Supplier Name
44	DRILLING_ENG	varchar(500)	Not Null	Drilling Engineer Name

2.6.2 Table Name: VMX_TIME_LOG

Description: This table stores header information of time logs
Primary Key: WELL_ID, WELLBORE_ID, LOG_ID
Foreign Key: WELL_ID, WELLBORE_ID

Sr. No.	Name	Datatype	Constraint	Description
1	WELL_ID	varchar(100)	Primary Key, Foreign Key	Well ID
2	WELLBORE_ID	varchar(100)	Primary Key, Foreign Key	Wellbore ID
3	LOG_ID	varchar(100)	Primary Key	Log ID
4	LOG_NAME	varchar(500)	Not Null	Log Name
5	RUN_NO	varchar(50)	Not Null	Run No
6	SERVICE_COMPANY	varchar(500)	Not Null	Service Company
7	COMMENTS	varchar(500)	Not Null	Time Log Comments
8	DATA_TABLE_NAME	varchar(500)	Not Null	Data Table Name
9	DESCRIPTION	varchar(500)	Not Null	Description
10	MAX_DATE	datetime	Not Null	Maximum Date
11	MAX_DEPTH	decimal(20, 7)	Not Null	Maximum Depth
12	MIN_DATE	datetime	Not Null	Minimum Date
13	MIN_DEPTH	decimal(20, 7)	Not Null	Minimum Depth
14	WMLS_URL	varchar(500)	Not Null	WMLS Server Address
15	WMLP_URL	varchar(500)	Not Null	WMLP Server Address

16	LAST_DATA_RECEIVED_ON	datetime	Not Null	Last Data Received On Date Time
17	CREATED_BY	varchar(50)	Not Null	Name of the user who created the record
18	CREATED_DATE	datetime	Not Null	Created Date
19	MODIFIED_BY	varchar(50)	Not Null	Name of the user who last modified the record
20	MODIFIED_DATE	datetime	Not Null	Modification Date
21	EDR_PROVIDER	varchar(255)	Not Null	EDR Provider Name
22	PRIMARY_LOG	numeric(1, 0)	Not Null	Is Primary Log : Yes, No
23	REMARKS_LOG	numeric(1, 0)	Not Null	Is Remarks Log : Yes, No
24	LAST_WRITE_DATE	datetime	Not Null	Date time of last record uploaded to the WITSML store
25	PI_WELL_ID	varchar(100)	Not Null	Pi Database Well ID
26	PI_WELLBORE_ID	varchar(100)	Not Null	Pi Database Wellbore ID
27	PI_LOG_ID	varchar(100)	Not Null	Pi Database Log ID

2.6.3 Table Name: VMX_DEPTH_LOG

Description: This table stores Depth Log header information.
Primary Key: WELL_ID, WELLBORE_ID, LOG_ID
Foreign Key: WELL_ID, WELLBORE_ID

Sr. No.	Name	Datatype	Constraint	Description
1	WELL_ID	varchar(100)	Primary Key, Foreign Key	Well ID
2	WELLBORE_ID	varchar(100)	Primary Key, Foreign Key	Wellbore ID
3	LOG_ID	varchar(100)	Primary Key	Depth Log ID
4	LOG_NAME	varchar(500)	Not Null	Depth Log Name
5	RUN_NO	varchar(50)	Not Null	Run No
6	SERVICE_COMPANY	varchar(500)	Not Null	Name of Service Company
7	COMMENTS	varchar(500)	Not Null	Comments on Depth Log
8	DATA_TABLE_NAME	varchar(500)	Not Null	Data Table Name of Depth Log
9	DESCRIPTION	varchar(500)	Not Null	Brief Description of Depth Log

10	MAX_DEPTH	decimal(20, 7)	Not Null	Maximum Depth
11	MIN_DEPTH	decimal(20, 7)	Not Null	Minimum Depth
12	WMLS_URL	varchar(500)	Not Null	WMLS Url
13	WMLP_URL	varchar(500)	Not Null	WMLP Url
14	LAST_DATA_RECEIVED_ON	datetime	Not Null	Last Data Received on (DateTime)
15	CREATED_BY	varchar(50)	Not Null	Name of the user who created the record
16	CREATED_DATE	datetime	Not Null	Created Date
17	MODIFIED_BY	varchar(50)	Not Null	Name of the user who last modified the record
18	MODIFIED_DATE	datetime	Not Null	Modification Date
19	M_DEPTH	decimal(20, 7)	Not Null	Max. Depth
20	EDR_PROVIDER	varchar(255)	Not Null	EDR Provider Name
21	LAST_WRITE_DEPTH	numeric(16,5)	Not Null	Depth of last record uploaded to the WITSML store
22	PI_WELL_ID	varchar(100)	Not Null	Pi Database Well ID
23	PI_WELLBORE_ID	varchar(100)	Not Null	Pi Database Wellbore ID
24	PI_LOG_ID	varchar(100)	Not Null	Pi Database Log ID

2.6.4 Table Name: VMX_TIME_LOG_COLUMNS

Description: This table stores information of time log channels.
Primary Key: WELL_ID, WELLBORE_ID, LOG_ID, MNEMONIC
Foreign Key: WELL_ID, WELLBORE_ID, LOG_ID

Sr. No.	Name	Datatype	Constraint	Description
1	WELL_ID	varchar(100)	Primary Key, Foreign Key	Well ID
2	WELLBORE_ID	varchar(100)	Primary Key, Foreign Key	Wellbore ID
3	LOG_ID	varchar(100)	Primary Key, Foreign Key	Log ID
4	MNEMONIC	varchar(100)	Primary Key	Channel Mnemonic
5	CHANNEL_NAME	varchar(500)	Not Null	Channel Name
6	DATA_TYPE	varchar(20)	Not Null	Data Type
7	UNIT	varchar(50)	Not Null	Channel Unit Description
8	UNIT_ID	varchar(50)	Not Null	Unit ID
9	VUMAX_UNIT_ID	varchar(50)	Not Null	VuMax Unit ID

10	VALUE_TYPE	decimal(1, 0)	Not Null	Value Type 0 - Static Value, 1 - Calculated Channel
11	VALUE_QUERY	varchar(4000)	Not Null	Calculation Expression
12	WITSML_MNEMONIC	varchar(100)	Not Null	WITSML Server Mnemonic
13	CREATED_BY	varchar(50)	Not Null	Name of the user who created the record
14	CREATED_DATE	datetime	Not Null	Created Date
15	MODIFIED_BY	varchar(50)	Not Null	Name of the user who last modified the record
16	MODIFIED_DATE	datetime	Not Null	Modification Date
17	COLUMN_ORDER	numeric(5, 0)	Not Null	Column Display Order
18	WRITE_BACK	numeric(1)	Not Null	Write back to the WITSML store?
19	OFFSET	numeric(16,5)	Not Null	Sensor Offset Value
20	NO_INTERPOLATE	numeric(1)	Not Null	Do not interpolate null values?
21	PI_MNEMONIC	varchar(100)	Not Null	Pi Database Mnemonic

2.6.5 Table Name: VMX_DEPTH_LOG_COLUMNS

Description: This table stores depth log channels information.
Primary Key: WELL_ID, WELLBORE_ID, LOG_ID, MNEMONIC
Foreign Key: WELL_ID, WELLBORE_ID, LOG_ID

Sr. No.	Name	Datatype	Constraint	Description
1	WELL_ID	varchar(100)	Primary Key, Foreign Key	Well ID
2	WELLBORE_ID	varchar(100)	Primary Key, Foreign Key	Wellbore ID
3	LOG_ID	varchar(100)	Primary Key, Foreign Key	Depth Log ID
4	MNEMONIC	varchar(100)	Primary Key	Channel ID
5	CHANNEL_NAME	varchar(500)	Not Null	Channel Name
6	DATA_TYPE	varchar(20)	Not Null	Data Type Eg. Double, Int. Etc...
7	UNIT	varchar(50)	Not Null	Unit Name
8	UNIT_ID	varchar(50)	Not Null	Unit ID
9	VUMAX_UNIT_ID	varchar(50)	Not Null	VuMax Unit ID
10	VALUE_TYPE	decimal(1, 0)	Not Null	Value Type, 0 - Static, 1 - Calculated Value

11	VALUE_QUERY	varchar(1000)	Not Null	Calculation Expression
12	WITSML_MNEMONIC	varchar(100)	Not Null	WITSML Mnemonic Name
13	CREATED_BY	varchar(50)	Not Null	Name of the user who created the record
14	CREATED_DATE	datetime	Not Null	Created Date
15	MODIFIED_BY	varchar(50)	Not Null	Name of the user who last modified the record
16	MODIFIED_DATE	datetime	Not Null	Modification Date
17	COLUMN_ORDER	numeric(5, 0)	Not Null	Column Display Order
18	WRITE_BACK	numeric(1)	Not Null	Write data back to WITSML store?
19	PI_MNEMONIC	varchar(100)	Not Null	Pi Database Mnemonic

2.6.6 Table Name: VMX_ALARM_PANEL

Description: This table stores well specific Alarm Panel data. This table is no longer used and was kept for backward compatibility. The Alarm Profile is used instead of alarm panel.

Primary Key: WELL_ID, PANEL_ID

Foreign Key: WELL_ID

Sr. No.	Name	Datatype	Constraint	Description
1	WELL_ID	varchar(100)	Primary Key, Foreign Key	Well ID
2	PANEL_ID	varchar(50)	Primary Key	Unique Panel ID, Auto Generated
3	PANEL_NAME	varchar(100)	Not Null	Alarm Panel Name
4	IS_ACTIVE	numeric(1, 0)	Not Null	Active Status
5	CREATED_BY	varchar(50)	Not Null	Name of the user who created the record
6	CREATED_DATE	datetime	Not Null	Creation Date
7	MODIFIED_BY	varchar(50)	Not Null	Name of the user who last modified the record
8	MODIFIED_DATE	datetime	Not Null	Modification Date

2.6.7 Table Name: VMX_WELLBORE
Description: This table stores wellbore header information
Primary Key: WELL_ID, WELLBORE_ID
Foreign Key: WELL_ID

Sr. No.	Name	Datatype	Constraint	Description
1	WELL_ID	varchar(100)	Primary Key, Foreign Key	Unique Well ID
2	WELLBORE_ID	varchar(100)	Primary Key	Wellbore ID
3	WELLBORE_NAME	varchar(500)	Not Null	Wellbore Name
4	WELLBORE_TYPE	varchar(100)	Not Null	Wellbore Type
5	WELLBORE_NO	varchar(100)	Not Null	Wellbore No.
6	GOVT_NO	varchar(100)	Not Null	Government No.
7	SHAPE	varchar(100)	Not Null	Wellbore Shape
8	STATUS	varchar(100)	Not Null	Wellbore Status : Completed/Drilling etc.
9	PURPOSE	varchar(100)	Not Null	Wellbore purpose
10	KICKOFF_DATE	datetime	Not Null	Wellbore Kick Off Date
11	DAY_TARGET	numeric(3, 0)	Not Null	Wellbore Day Target
12	MD_CURRENT	numeric(16, 5)	Not Null	Current Measured Depth
13	TVD_CURRENT	numeric(16, 5)	Not Null	Current True Vertical Depth
14	MD_KICKOFF	numeric(16, 5)	Not Null	Measured Kick Off Value
15	TVD_KICKOFF	numeric(16, 5)	Not Null	True Vertical Kick Off Value
16	MD_PLANNED	numeric(16, 5)	Not Null	Planned Measured Depth
17	TVD_PLANNED	numeric(16, 5)	Not Null	Planned True Vertical Depth
18	MD_SS_PLANNED	numeric(16, 5)	Not Null	Planned MD SubSurface
19	TVD_SS_PLANNED	numeric(16, 5)	Not Null	Planned TVD SubSurface
20	WMLS_URL	varchar(500)	Not Null	WITSML Server Address
21	WMLP_URL	varchar(500)	Not Null	WMPL Server Address
22	CREATED_BY	varchar(50)		Name of the user who created the record
23	CREATED_DATE	datetime		Created Date
24	MODIFIED_BY	varchar(50)		Name of the user who last modified the record
25	MODIFIED_DATE	datetime		Modification Date

2.6.8 Table Name: VMX_WELL_COLUMNS

Description: This table stores master information of well columns to display in the Data Service dashboard

Sr. No.	Name	Datatype	Constraint	Description
1	COLUMN_ID	varchar(100)	Not Null	Well Column ID
2	VISIBLE	numeric(1, 0)	Not Null	Well Column Visible : Yes / No

2.6.9 Table Name: VMX_USER

Description: This table stores information of VuMaxDR users created through user manager

Primary Key: USER_ID

Sr. No.	Name	Datatype	Constraint	Description
1	CREATED_BY	varchar(50)	Not Null	Name of the user who created the record
2	CREATED_DATE	datetime	Not Null	Created Date
3	MODIFIED_BY	varchar(50)	Not Null	Name of the user who last modified the record
4	MODIFIED_DATE	datetime	Not Null	Modification Date
5	PASSWORD	varchar(100)	Not Null	User Password
6	REMARKS	varchar(500)	Not Null	User Remarks
7	USER_ID	decimal(5, 0)	Primary Key	User ID
8	USER_NAME	varchar(50)	Not Null	User Name