Design Document

For

Urban Service System,

a web portal of service provider system.

Prepared by

Debopriyo Ghosh Roll- 33

Saikat Jana Roll-50

Mousumi Mondal Roll-40

Saradindu Rana Roll-28

Srila Parui Roll-38

B.Tech., Sem - V, Group- 1

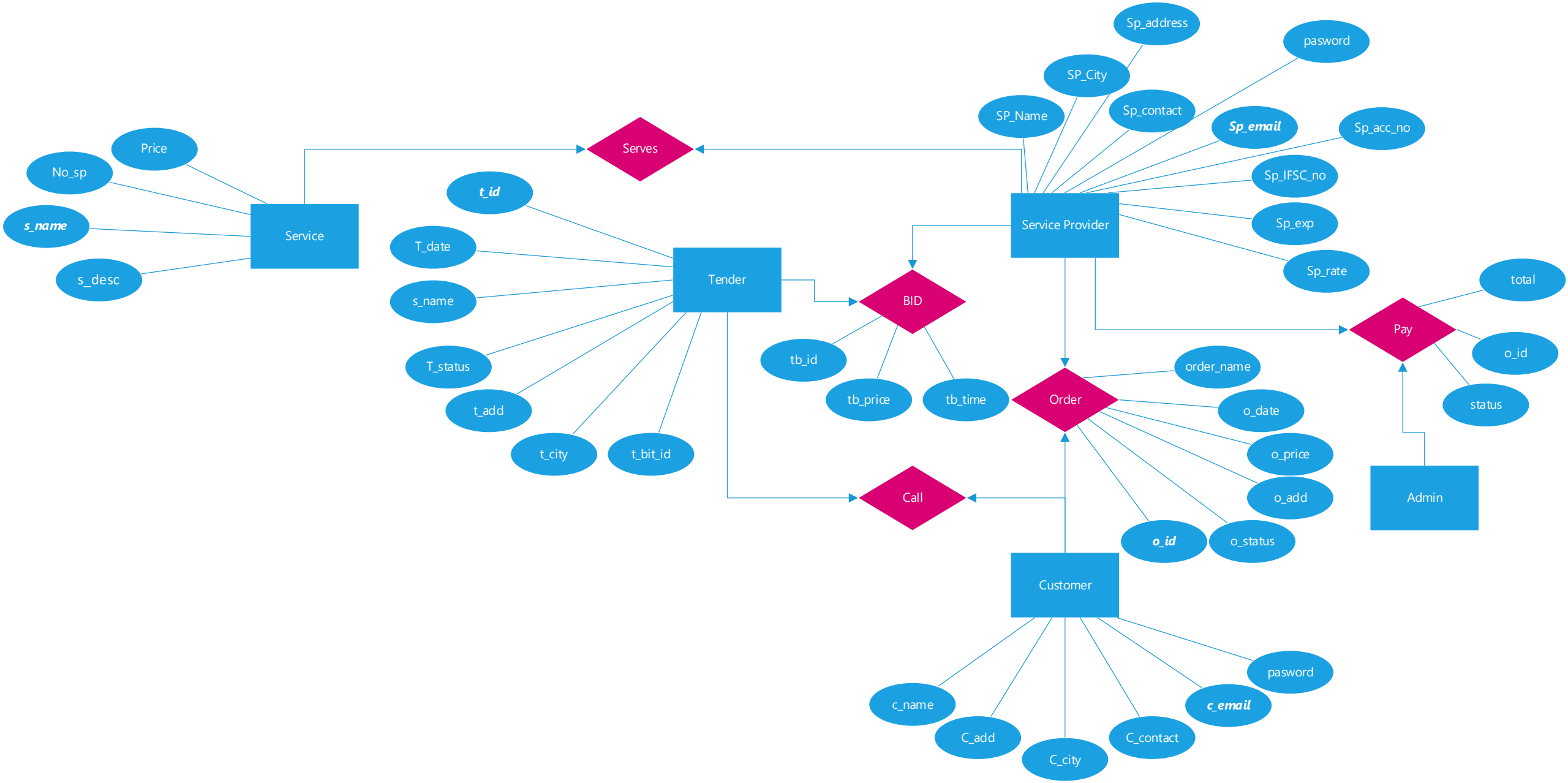
Dated: 07.03.2021

Contents

Entity-Relationship Diagram

An **entity–relationship model** (or **ER model**) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database, typically a relational database.

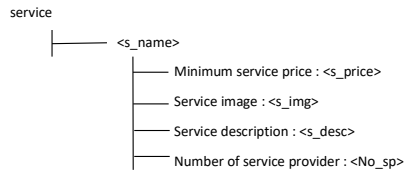

Sp_address
pasword
SP_City
SP_Name
Sp_contact
Sp_email
Sp_acc_no
Sp_IFSC_no
Sp_exp
Sp_rate
Price
No_sp
s_name
s_desc
Service
Serves
Service Provider
t_id
T_date
s_name
T_status
t_add
t_city
t_bit_id
Tender
BID
tb_id
tb_price
tb_time
Pay
total
o_id
status
Admin
order_name
o_date
o_price
o_add
o_id
o_status
Order
Call
Customer
pasword
c_name
C_add
C_city
C_contact
c_email

Relational Schema

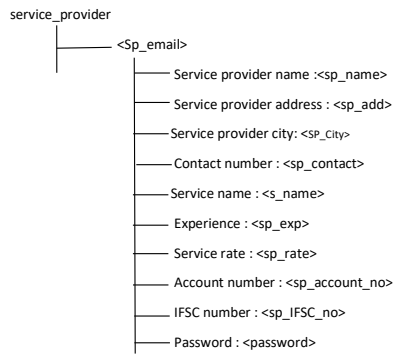
Relation schema defines the design and structure of the relation like it consists of the relation name, set of attributes/field names/column names. every attribute would have an associated domain.

A **relational schema** is a set of **relational** tables and associated items that are related to one another. All of the base tables, views, indexes, domains, user roles, stored modules, and other items that a user creates to fulfill the data needs of a particular enterprise or set of applications belong to one **schema**.

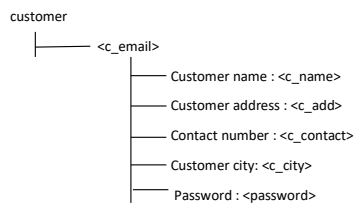
Service:



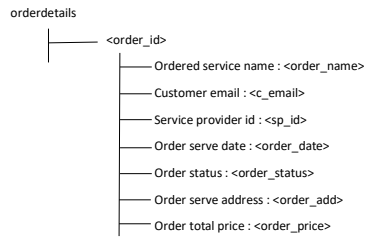
Service Provider:



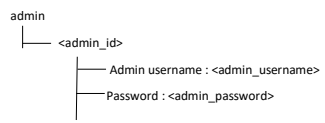
Customer:



Order Details:



Admin:



Data Dictionary

A **data dictionary** contains metadata i.e **data** about the database. The **data dictionary** is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc. . Physical information about the tables such as where they are stored and how.

Table	Column	Data Type	Primary key	Nullable	Default	Comments
admin	admin_id	int(10)	YES	No		Admin's valid email ID
admin	admin_username	varchar(500)		No		Admin's user_name
admin	admin_password	varchar(500)		No		A valid password given by Admin

Table	Column	Data Type	Primary key	Nullable	Default	Comments
customer	c_name	text		No		Customer’s full name
customer	c_add	text		No		Customer’s address
customer	c_city	text		No		Customer’s city
customer	c_contact	varchar(13)		No		Customer’s contact number
customer	c_email	varchar(50)	YES	No		Customer’s valid email ID
customer	password	varchar(15)		No		Valid password given by Customer

Table	Column	Data Type	Primary key	Nullable	Default	Comments
service	s_name	varchar(50)	YES	No		Service name
service	s_price	int(10)		Yes	NULL	Service minimum price
service	s_desc	text		No		Service description
service	s_img	varchar(255)		Yes	NULL	Service demo images
service	No_sp	int(11)		No	0	Number of service provider in service

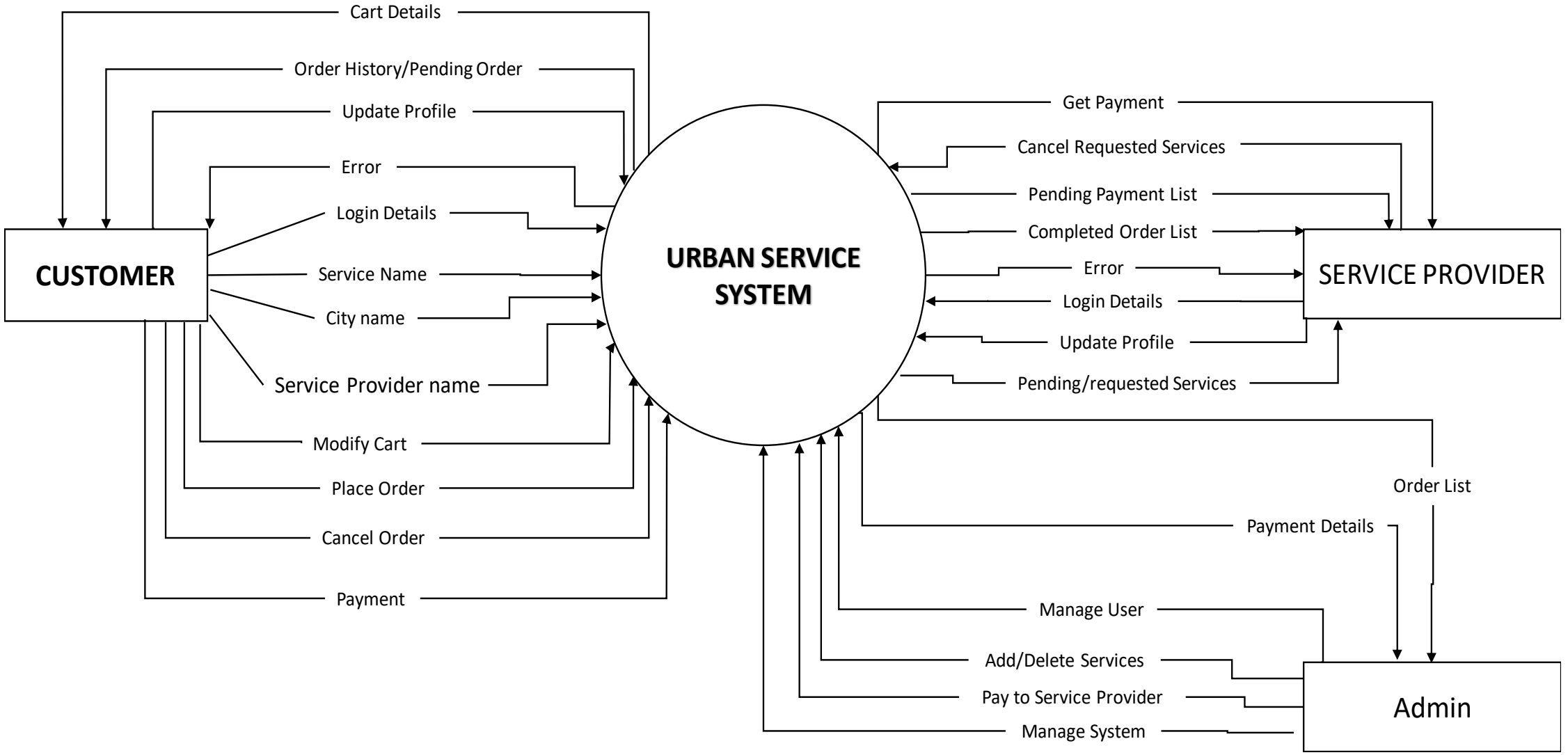
Table	Column	Data Type	Primary key	Nullable	Default	Comments
service_provider	sp_name	text		No		Name of service provider
service_provider	sp_add	text		No		Address of service provider
service_provider	sp_contact	varchar(13)		No		Cantact number of service provider
service_provider	sp_email	varchar(50)	YES	No		Valid email ID of a service provider
service_provider	sp_exp	int(11)		No		Year of experience of a service provider
service_provider	sp_rate	int(11)		No		Rate of a service provider for a service
service_provider	sp_account_no	varchar(20)		No		Service provider’s bank account number
service_provider	sp_IFSC_no	varchar(11)		No		Service provider’s bank account’s IFSCnumber
service_provider	sp_city	text		No		Service provider’s service city
service_provider	s_name	varchar(50)		No		Service name
service_provider	password	varchar(255)		No		Valid password given by service provider

Table	Column	Data Type	Primary Key	Nullable	Default	Comments
orderdetails	order_id	int(10)	YES	No		Unique id for a particular order
orderdetails	order_name	varchar(50)		No		Order service name
orderdetails	c_email	varchar(50)		No	0	Customer's email Id for a particular order
orderdetails	sp_email	varchar(50)		No		Service provider's email Id for a particular order
orderdetails	order_add	varchar(255)		No		Order serve address
orderdetails	order_price	double		No	0	Price for a particular order
orderdetails	order_status	varchar(45)		No		Order status for a particular order
orderdetails	order_date	date		No	0000-00-00	Order delivery date for a particular order

Context Diagram

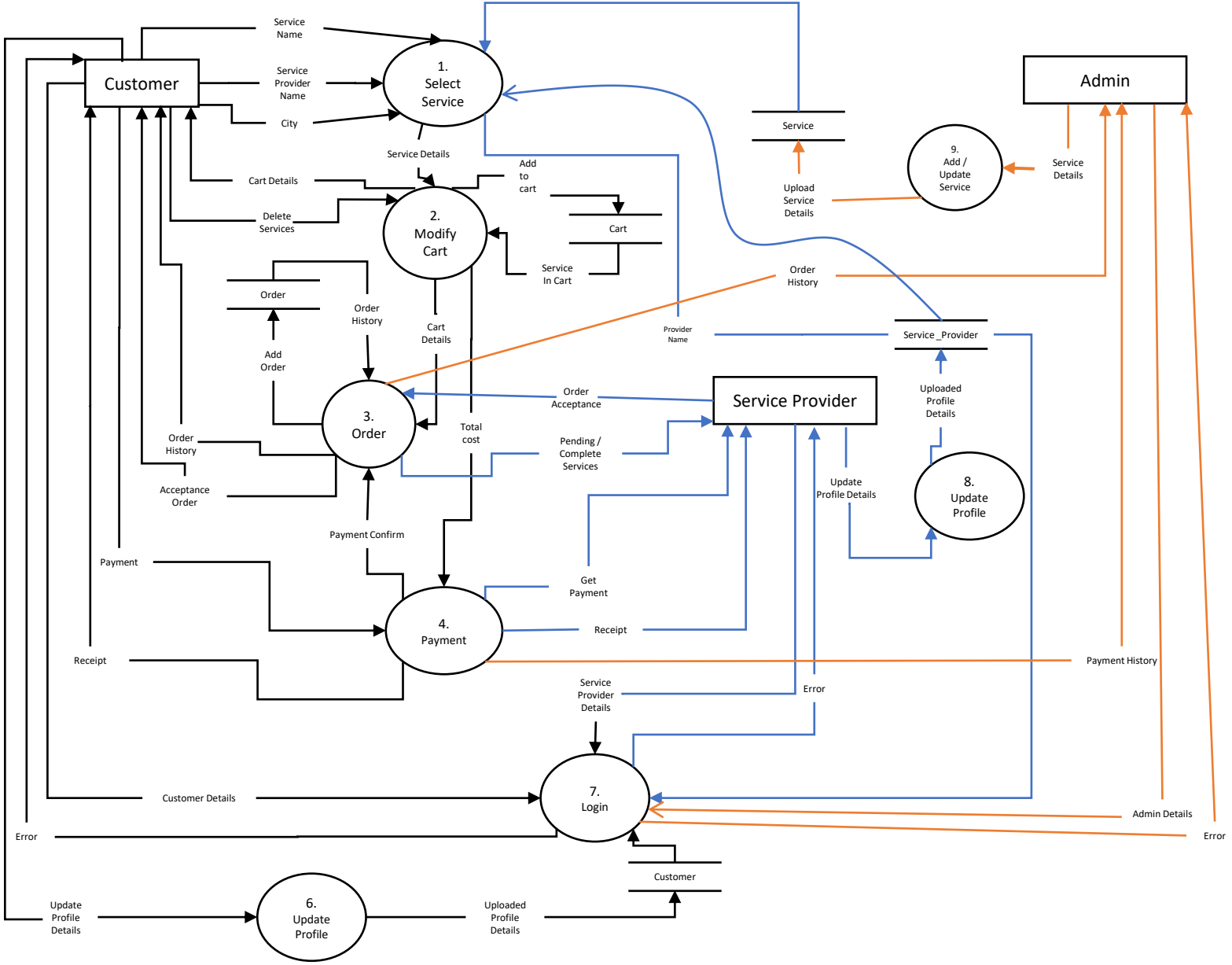
A **context diagram**, sometimes called a level 0 data-flow **diagram**, is drawn in order to define and clarify the boundaries of the software system. It identifies the flows of information between the system and external entities. The entire software system is shown as a single process.

Context Diagram



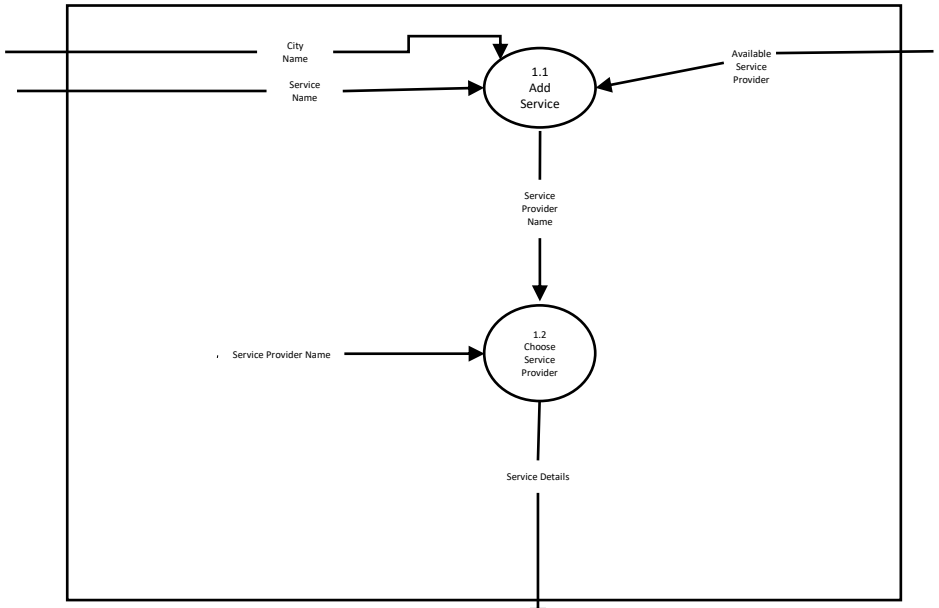
Data Flow Diagram

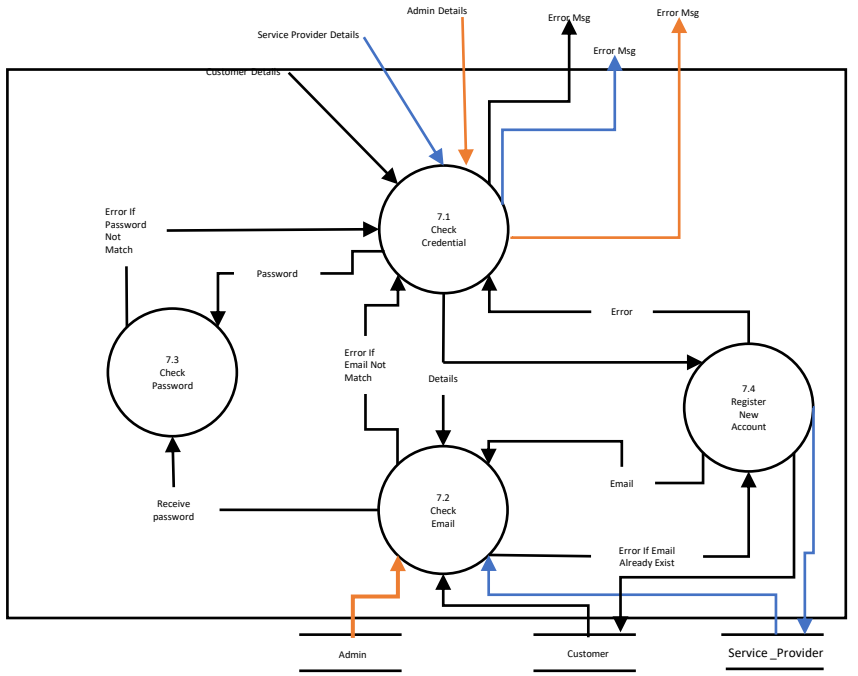
A **data**-**flow diagram** is a way of representing a **flow** of **data** through a **process** or a system (usually an information system). The **DFD** also provides information about the outputs and inputs of each entity and the **process** itself. Specific operations based on the **data** can be represented by a **flowchart**.



Level 1

Level 2





Data Flow Diagram