LONDON METROPOLITAN UNIVERSITY

ITAHARI INTERNATIONAL COLLEGE

Module Code & Module Title:

CS5002NT Software Engineering

Assessment Weightage & Type:

20% Group Coursework

Title:

Sound Strong Music Institute

Year and Semester:

2020-21 Autumn

Group Name: (Code Breakers)			
SN	**Student Name**	**College ID**	**University ID**
1.	**Ashmina Rai**	**NP05CP4A190033**	**19031606**
2.	**Ashesh Rai**	**NP05CP4A190031**	**19031902**
3.	**Bibek Pradhan**	**NP05CP4A190040**	**19031617**
4.	**Nishan Kadel**	**NP05CP4A190099**	**19031776**
5.	**Sachin Dahal**	**NP05CP4A190140**	**19031839**

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgement

We owe a great deal to express our special and heartfelt gratitude to Mr. Mission Sapkota Sir, our module lecturer, for his great support and humble guidance to maintain our confidence throughout the project. Our completion of this project could not have been accomplished this much better without the continuous support of him.

We are also very grateful towards the Itahari International College, offering our sincere appreciation for providing us an opportunity to develop and improve our skills via this productive assignment. This assignment was fruitful for all the members in our team. We got to know a lot of new things.

We are also thankful towards our seniors and friends who helped us somehow finalize the project.

Lastly, we would like to thank everybody who is associated with this assignment directly or indirectly, our module lecturer for his continuous support by which we are able to complete this assignment within the given time frame.

With best regards,
Ashesh Rai
Ashmina Rai
Bibek Pradhan
Nishan Kadel
Sachin Dahal

Table of Contents

Table of Figures

Table of Tables

1. Introduction

Here, in the given group coursework we must demonstrate the practical knowledge of “Structured Software Engineering” (Yourdon) that consist of total of 20% of an overall module weight. This report mainly consists the system requirement for “Sound Strong” which is a music institute and provides musical instruments training and recording service for a decade. It also provides a practice room for musicians as a rental service. But, recently its practice room service has been in demand quite lately with quality rooms for which it is facing difficulties to manage the heavy influx of customers and phone call-based booking. So, an online system for practice rooms/instruments hiring related processes is to be developed. Similarly, it also provides a discount scheme and special packages for the customers who have premium membership and also sends an automatic message before the expiry deadline so that they can renew their membership or stay as a customer.

Thus, the main objective of this coursework is to create the system for “Sound Strong” and meet all the assumption and required requirements like, to design ERD, Context level diagram, DFD and many more with the help of group members at a given period of time.

2. Symbol

Symbol is any signs, word or a number that gives a simple idea or an information. It can be used for various purpose as it makes people go beyond their thinking of what is seen or known creating a linkage. Here, in this coursework we also used different symbols which have their own meanings.

2.1 Data Flow Diagram Symbol

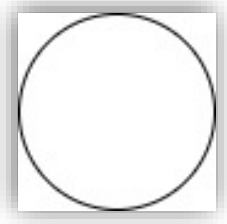
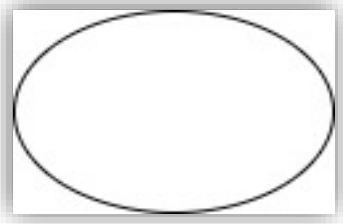
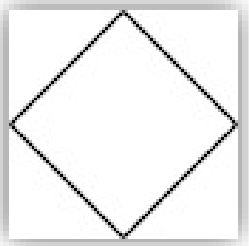
Symbol	Name	Meaning
	Data flow	It is used to connect processes with each other and its arrow head represents the direction of data flow.
	Data process	It performs transformation of input data to process output data.
	External Entity	It represents as a source of system inputs or a sink of system outputs.
	Data store	It is a repository of data where the arrow heads indicate net inputs where and net outputs to store.

Table 1: Symbol of DFD

2.2 ER Diagram Symbol

Symbol	Meaning
	It represents as entity.
	It represents as attribute.
	It represents as relation.
	It links attribute(s) to entity set(s) or entity set(s) to relationship set(s).
	It represents as a weak entity.
	It represents as multivalued attribute.

	It represents as a weak attribute.
	It represents as a key attribute or a single valued attribute.
	It represents as one to one relation.
	It represents as one to many relations.
	It represents as many to many relations.

Table 2: Symbol of ERD

2.3 Structure Chart Symbol

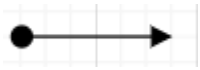
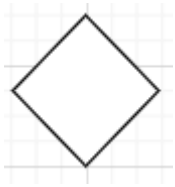
Symbol	Description
	**Module**: This symbol represents the process of certain functions.
	**Flag**: This symbol represents the control coupling that one module controls the flow of another by passing information on what to do.
	**Data Couple**: This symbol represents the data passed between the modules.
	Connector: This symbol is used to show the connections between modules.
	**Storage:** Storage is where all the information is to be stored.
	**Conditional Call:** It represents that control module can select any of the sub module on the basis of some condition.

Table 3: Symbol of Structure Chart

3. Group Task

3.1 Environmental Model Specification

3.1.1 Statement of Purpose

We are very charmed towards the project of a music institute, "Sound Strong". Firstly, we addressed all the requirements to move on for the further process towards the development of a product. Moving onwards to the requirements gathering process, we also inspected the flow and the working mechanism of the institute and proposed a simple idea of it. The events are also proposed for the system to have a continuous data flow between the users, staff, company-admin and the institute. The majority of data which is passed by the users will be stored in a database for the further usage by the company to manipulate or even use it in an upcoming future. Also, a working roadmap is created by the cross-functional team in an appropriate way for the systematic development of the project till the end.

Categorized external entities for the Sound Strong music institute are Admin, Staff, Customer and Member. There will be an admin for a company as a head to manage the staff with an ability to register and deregister the company staff. To hire any musical instruments or to book a room, a customer needs to be registered and also had to be accepted by the company staff. Customers can also get a membership tag from a company which is planned by the company on a subscription basis (which is recommended one too). If a new customer is in, to draft any musical instruments or even a room, he/she needs to pay some amount as an advance to the company up to their 8th booking if they haven't purchased the membership subscription package. Membership is planned to ensure a reliability between the customer and a company.

- Admin can have control of data of a company staff by which he/she can register or deregister the staff.
- Should be an asynchronous dataflow between the external entities and processes.
- Company staff will be maintaining all the records of booked or available rooms and instruments.
- Customer can send the data like paying fees/advance, looking for available rooms/instruments, also booking and cancelling rooms/instruments

- Every data passed by the customer will first go to the company staff where their request can be accepted or declined.
- Any customer can take a membership tag offered by the company as a membership package.
- Customers with membership will get notified about their membership status.
- Customers who purchased membership of a company will get a special discount.
- Staff can generate reports of each individual customer about all their previous activities in a system.
- Every detail of a customer's user and payment details are managed by the company staff.

3.1.2 Context Level Diagram

The Context diagram is often referred to as the Level 0 Data Flow Diagram in a Data Flow Diagram and is the highest level. It is a useful element among business analysts who use it to understand the system's specifics and boundaries to be established in a project. The flow of information between the system and external components is illustrated.

It is composed of a bubble of background, first drawn in the center of the map. Typically, it is a circle shape that reflects a conceptual limit that involves a collective of a project's interconnected processes and activities. As it is purely a high-level view of the system, the nitty-gritty specifics of a system's internal structure are masked in a background diagram. This process is called hiding information.

In a project, a context diagram forms part of the specifications document. The context diagram, unlike other project diagrams, is not for the engineers/technicians to use, but for the stakeholders of the project. It should also be addressed in clear and comprehensible language to make it easier for stakeholders to understand things while evaluating them. (Wondershare EdrawMax, 2020)

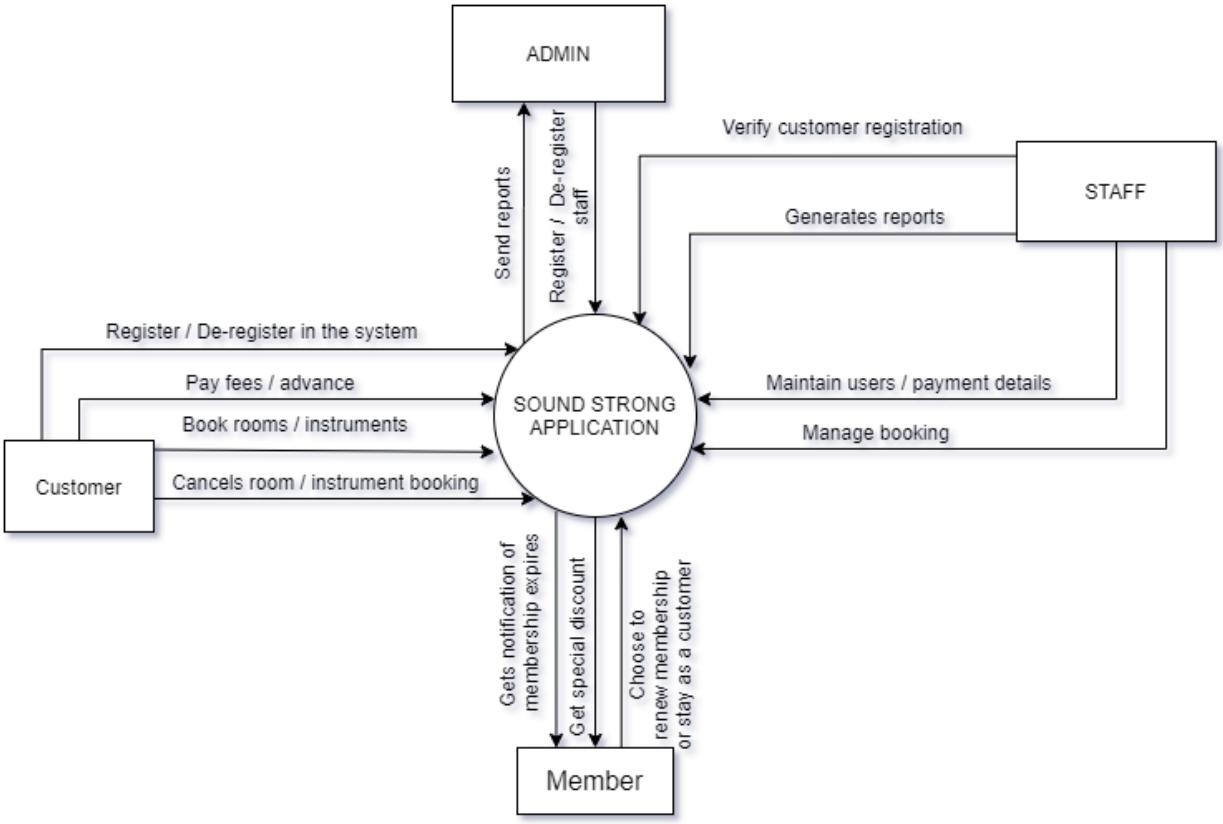


Figure 1: Context Level Diagram

The figure above is the Sound Strong application's context level diagram. Generally, the context diagram connects our system's three entities to the application. In the above background diagram, clients, workers and administrators are connected to the Sound Strong application framework where the data flows at the system surface level are seen.

3.1.3 Event List

Events are objects or messages used when a part of the program needs to notify other elements of a change of state.

An Event model is a program (a series of classes and interfaces) architecture that specifies how components can:

On the Origin Case side:

- Event formation and definition
- Event Cause (or Fire)
- Distribute incidents to the elements involved

On the listeners' side of the event:

- Subscribe to sources for incidents
- When obtained, react to events
- Where desired, delete the subscription to event sources

List of Event

1. Admin register staff (flow-oriented event),
2. Admin de-register staff (flow-oriented event),
3. Customer registers in the system (flow-oriented event),
4. Customer registers as a member (flow-oriented event),
5. Customer deregister from the system (flow-oriented event),
6. Customer re-active membership (temporal event),
7. Staff generates reports (temporal event),
8. Staff assigns instruments (flow-oriented event),
9. Admin gets reports from system (temporal event),
10. Staff manages booking (flow-oriented event),
11. Staff verifies customer registration (flow-oriented event),
12. Staff maintains user's payment (flow-oriented event),
13. Member gets special discount (flow-oriented event),
14. Member gets notified of membership expires before a week (temporal event),
15. Member chooses to renew membership (flow-oriented event),
16. Member cancels membership (flow-oriented event),
17. Customer pays fee (flow-oriented event),
18. Customer pays advance (flow-oriented event),
19. Customer books room (temporal event),
20. Customer books instruments (temporal event),
21. Customer cancels room booking (flow-oriented event),
22. Customer cancels instruments booking (flow-oriented event).

3.1.4 DFD Fragmentation

A DFD fragment is created for each use case triggered by an event in the event table. Each DFD fragment is a self-contained model showing how the system responds to a single event. The analyst usually creates DFD fragments one at a time, focusing attention on each part of the system. The DFD fragments are drawn after the event table and context diagram are complete (Coursehero, 2020).

- DFD fragments for “Admin register staff.”

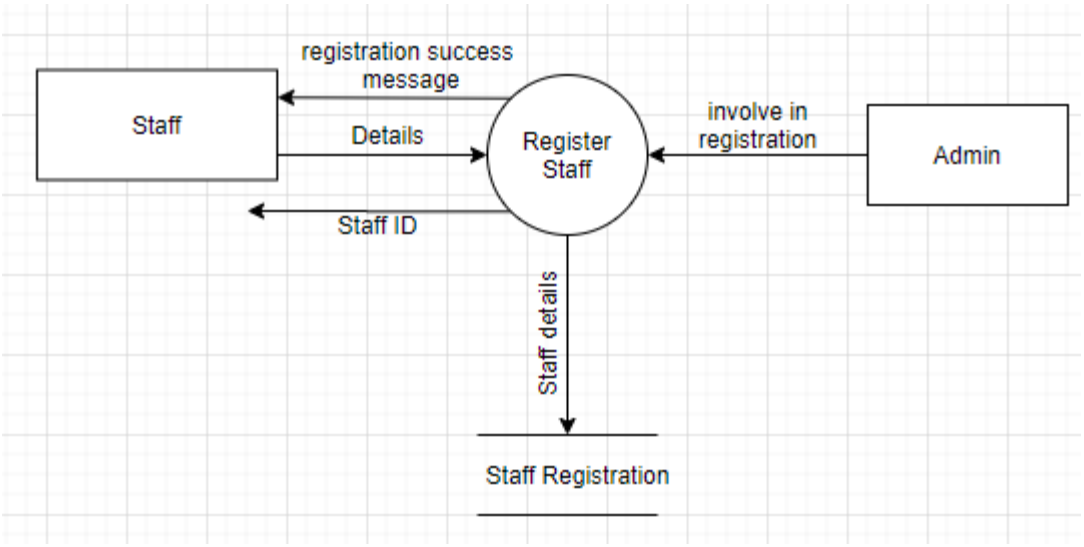


Figure 2: Admin Register Staff

- DFD fragments for “Admin de-register staff.”

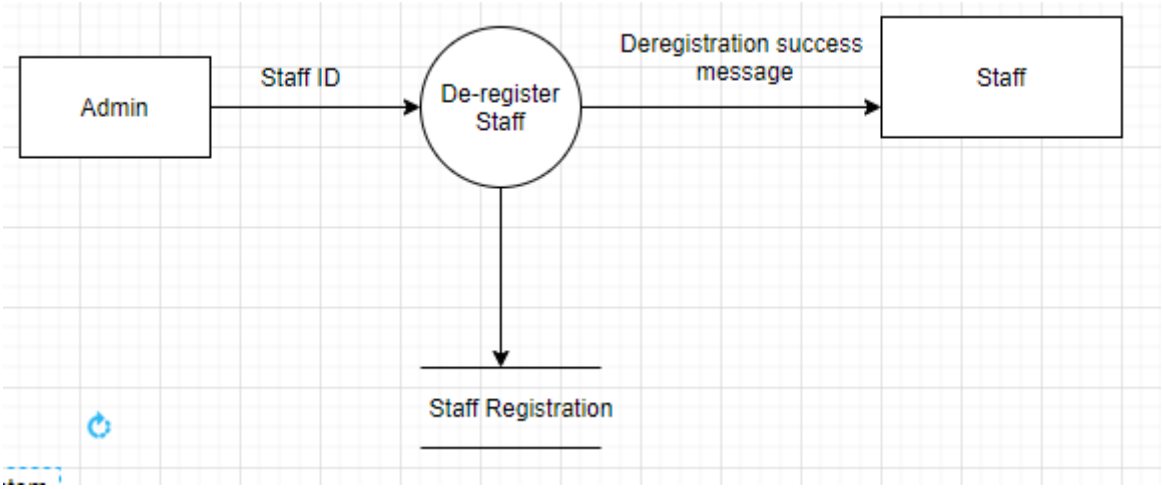


Figure 3: Admin De-register Staff

- DFD fragments for “Customer registers in the system.”

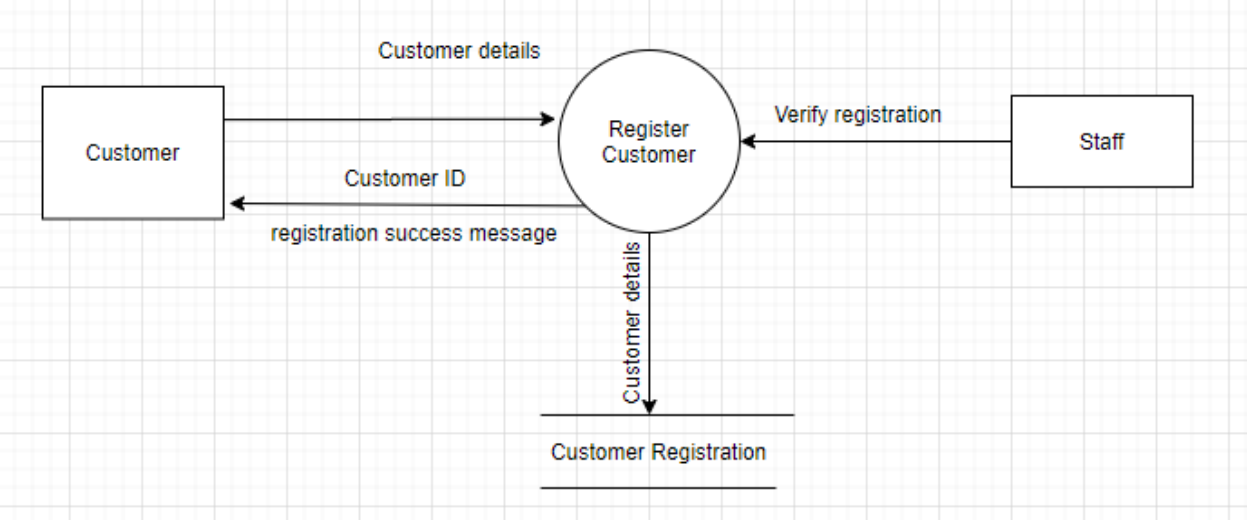


Figure 4: Customer Registers in the System

- DFD fragments for “Customer registers as a member.”

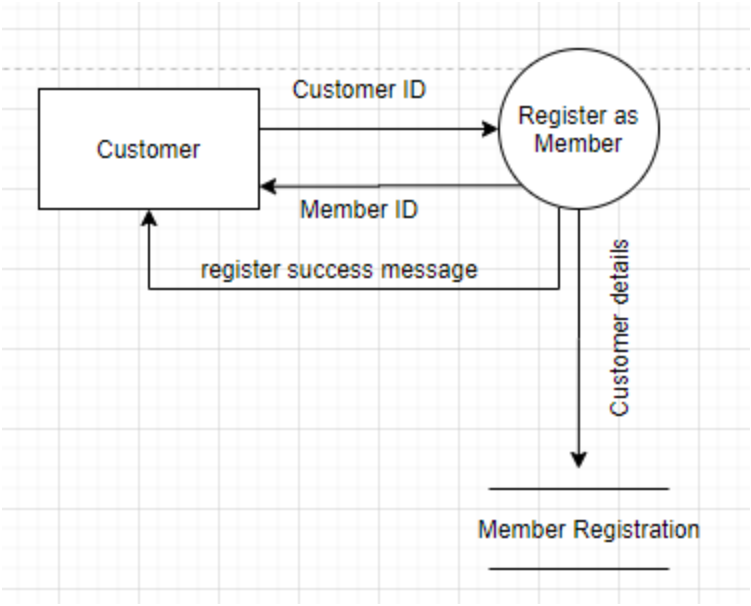


Figure 5: Customer Registers as a Member

- DFD fragments for “Customer deregister from the system”

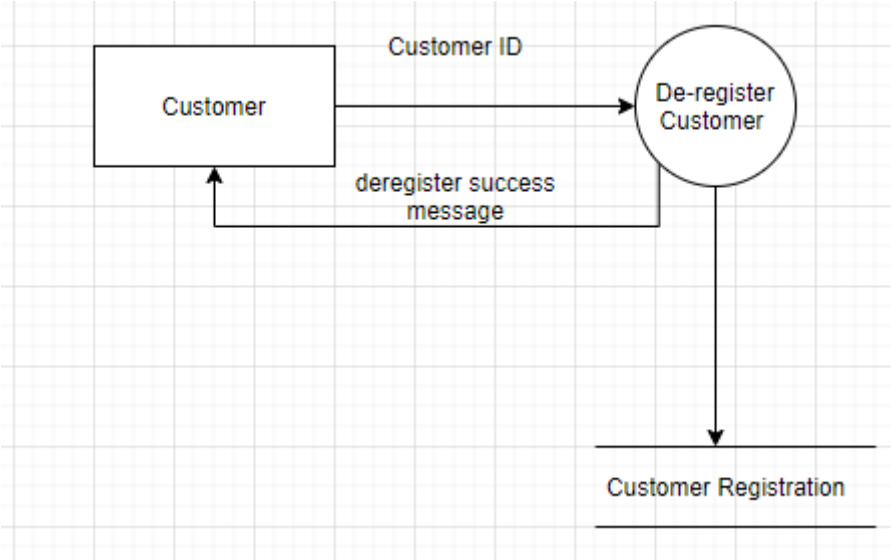


Figure 6: Customer De-register from the System

- DFD fragments for “Customer re-active membership”

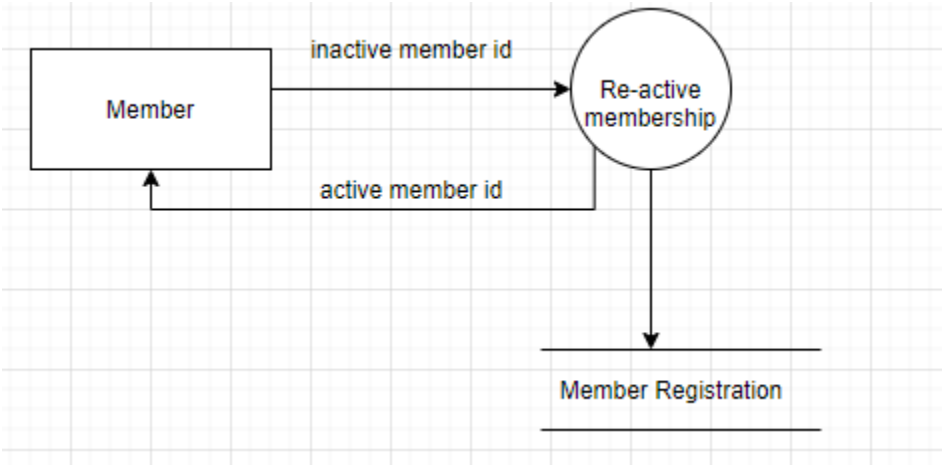


Figure 7: Customer Re-active Membership

- DFD fragments for “Staff generates reports”

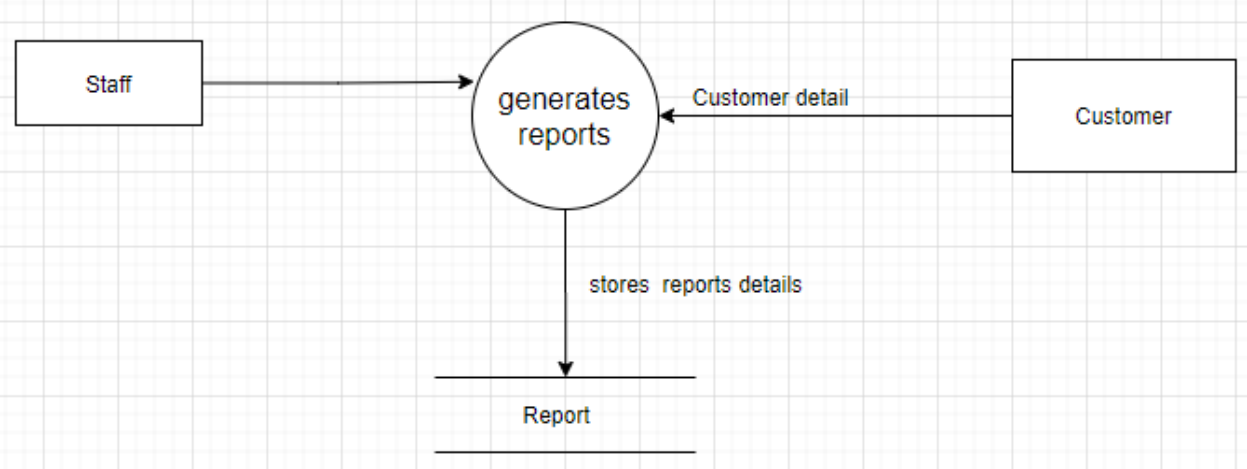


Figure 8: Staff Generates Reports

- DFD fragments for “Staff assigns instruments”.

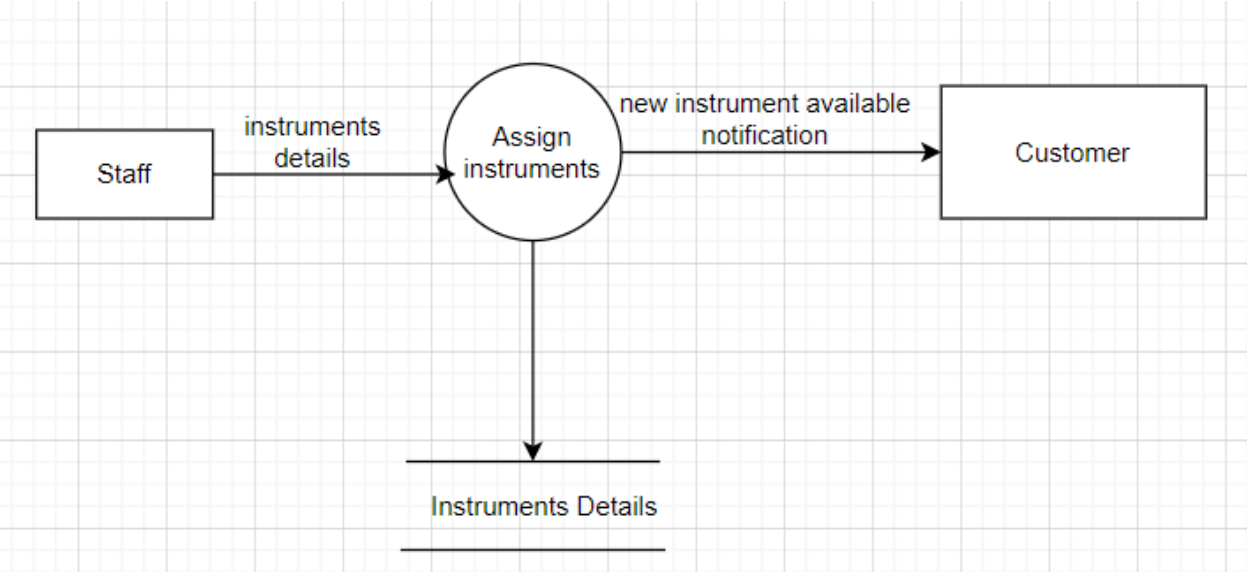


Figure 9: Staff Assigns Instruments

- DFD fragments for “Admin gets reports from system.”

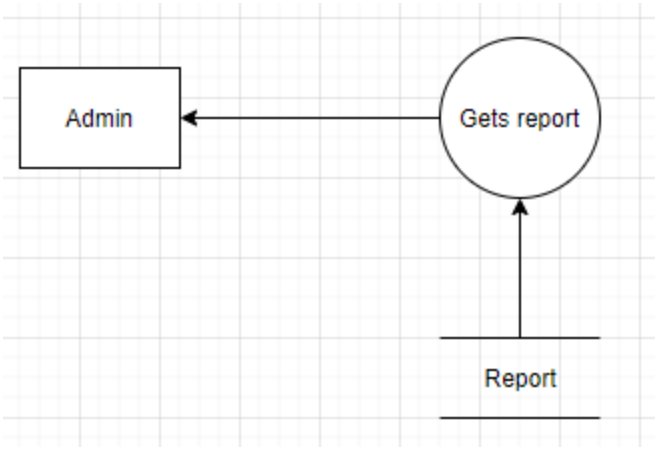


Figure 10: Admin gets Reports from System

- DFD fragments for “Staff manages booking.”

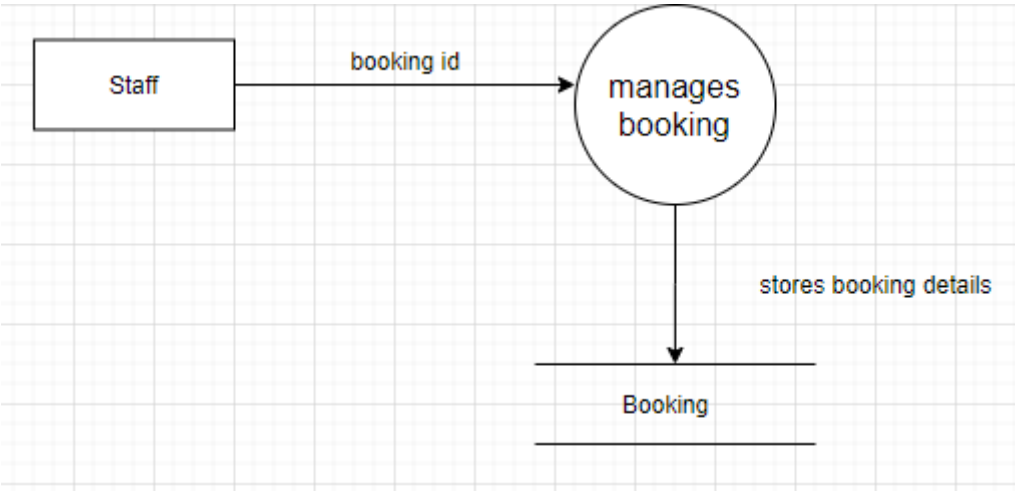


Figure 11: Staff Manages Booking

- DFD fragments for “Staff verifies customer registration”

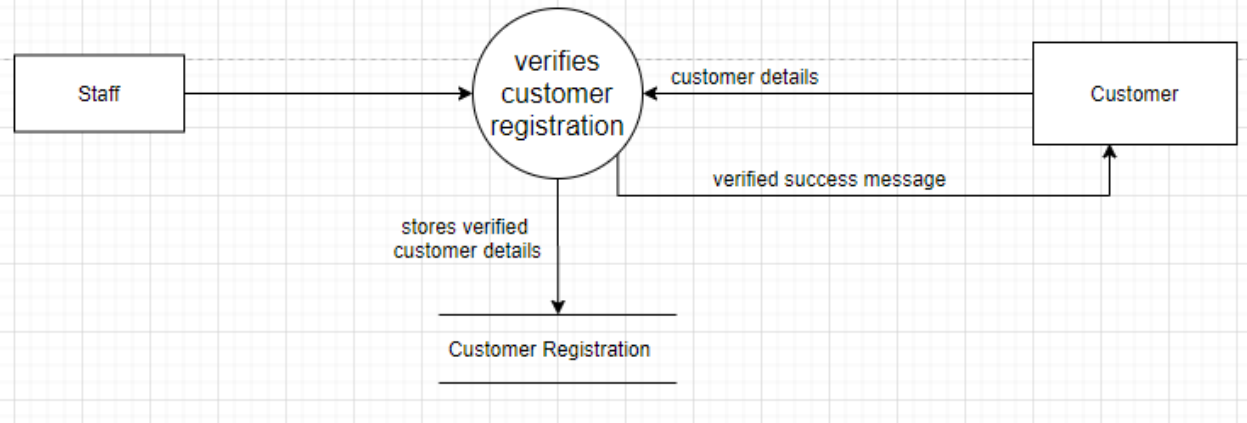


Figure 12: Staff Verifies Customer Registration

- DFD fragments for “Staff maintains user’s payment.”

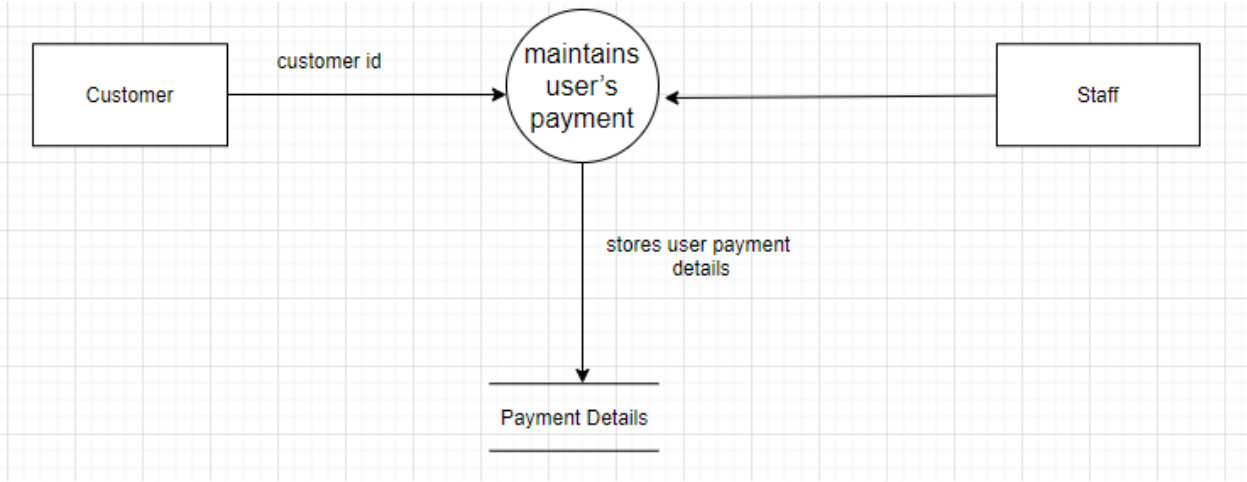


Figure 13: Staff Maintains User's Payment

- DFD fragments for “Member gets notified of membership expires before a week.”

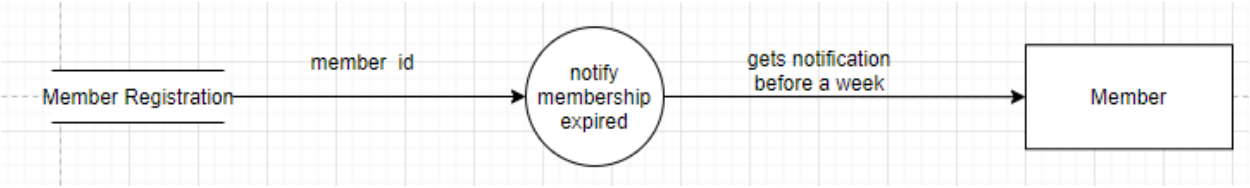


Figure 14: Member gets Notified of Membership Expires Before a Week

- DFD fragments for “Member chooses to renew membership.”

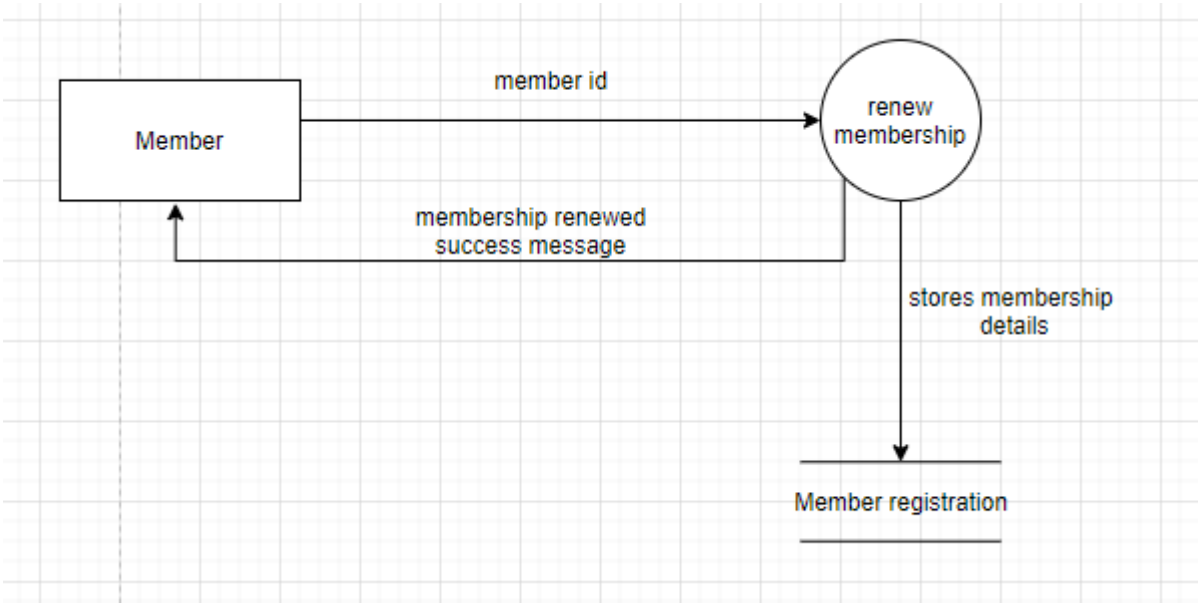


Figure 15: Member Chooses to Renew Membership

- DFD fragments for “Member cancels membership.”

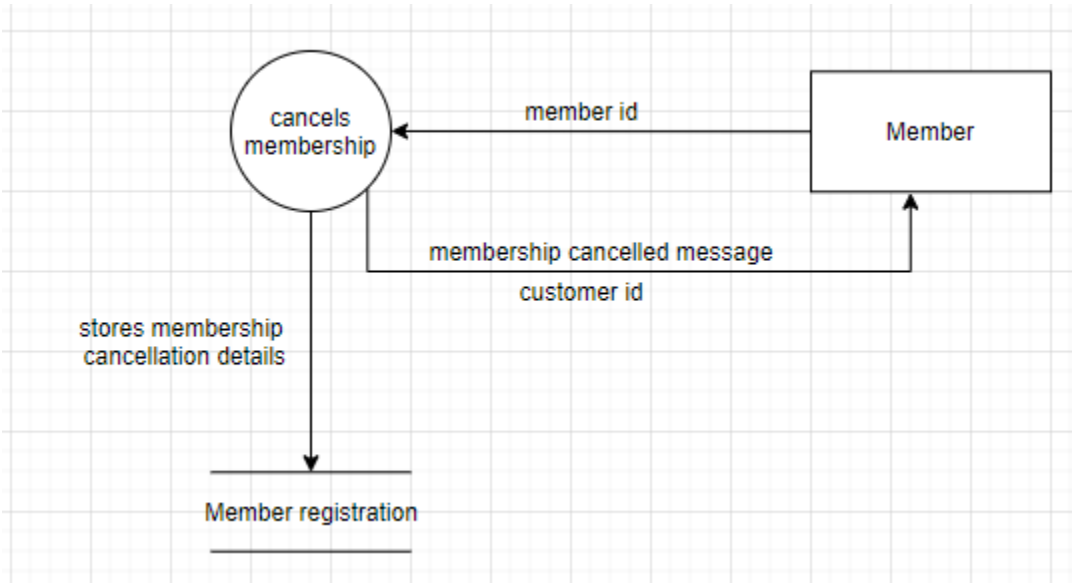


Figure 16: Member Cancels Membership

- DFD fragments for “Customer pays fee.”

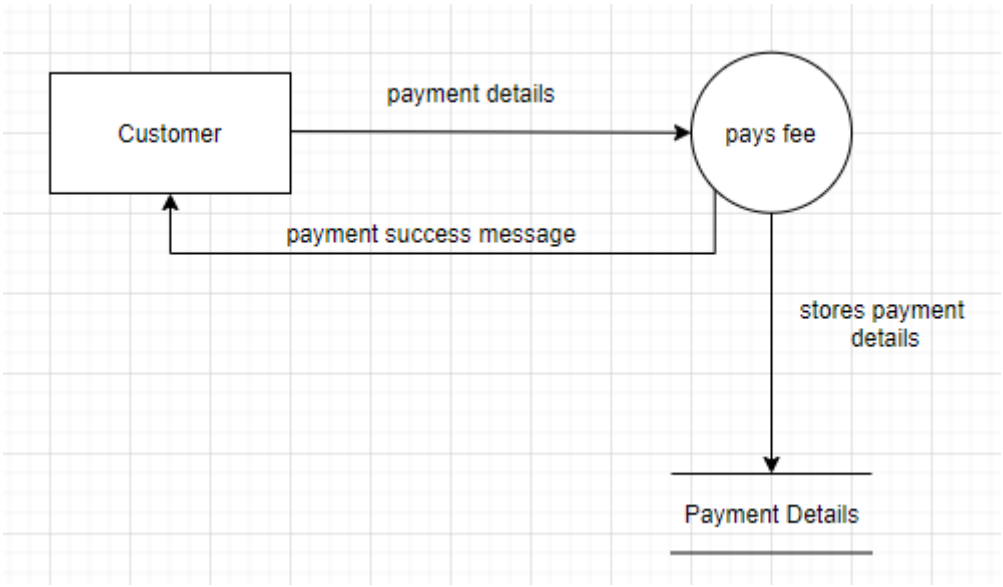


Figure 17: Customer Pays Fee

- DFD fragments for “Customer pays advance.”

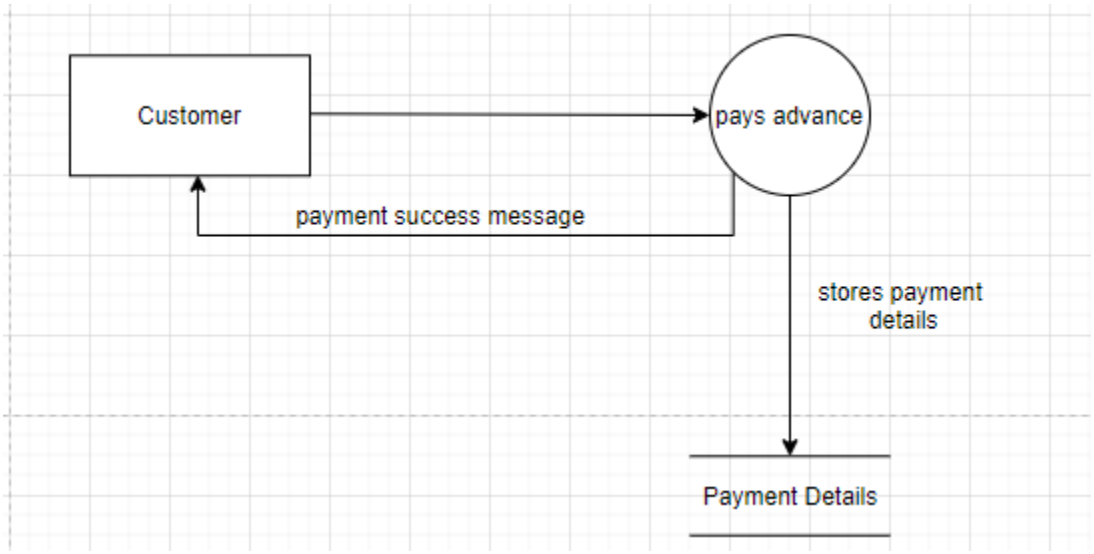


Figure 18: Customer Pays Advance

- DFD fragments for “Customer books room.”

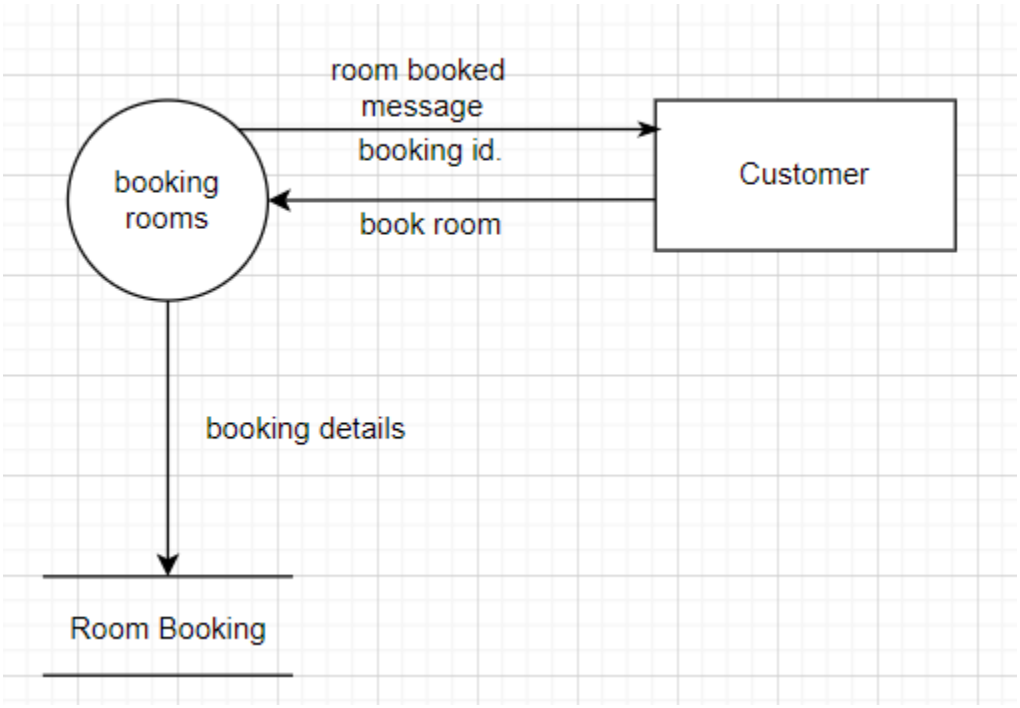


Figure 19: Customer Books Room

- DFD fragments for “Customer books instrument.”

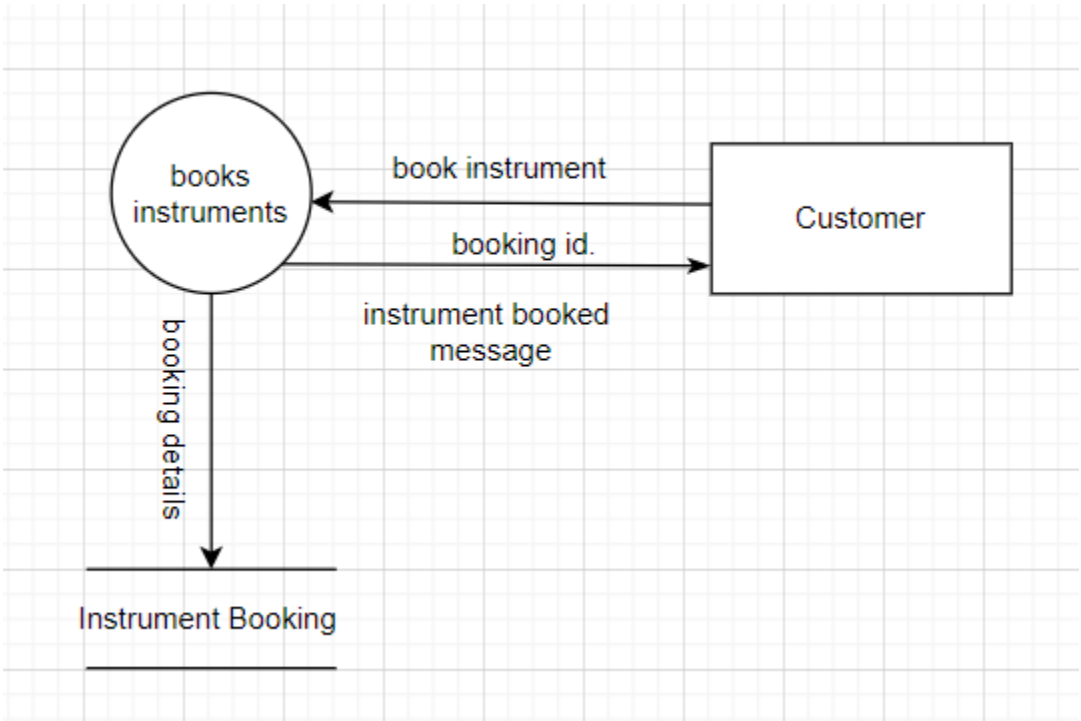


Figure 20: Customer Book Instruments

- DFD fragments for “Customer cancels room booking”

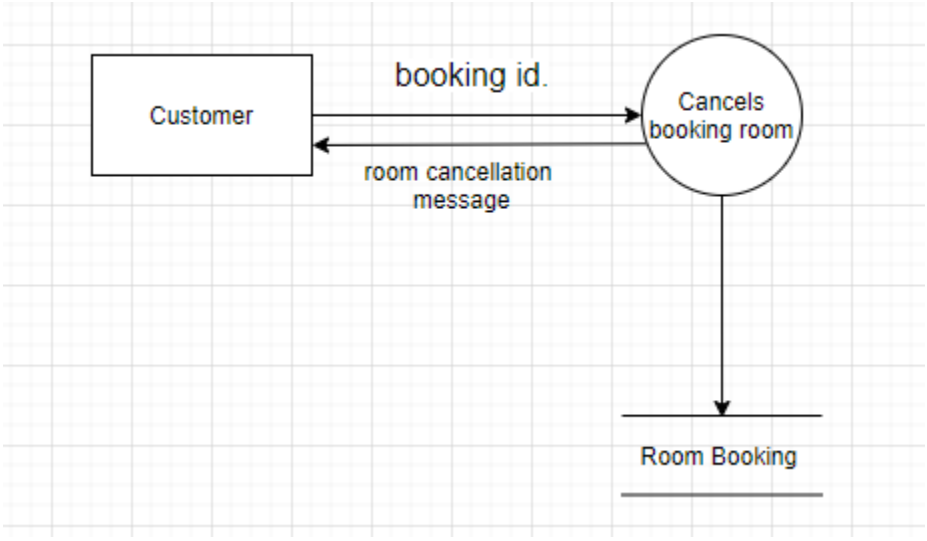


Figure 21: Customer Cancels Room Booking

- DFD fragments for “Customer cancels instruments booking”

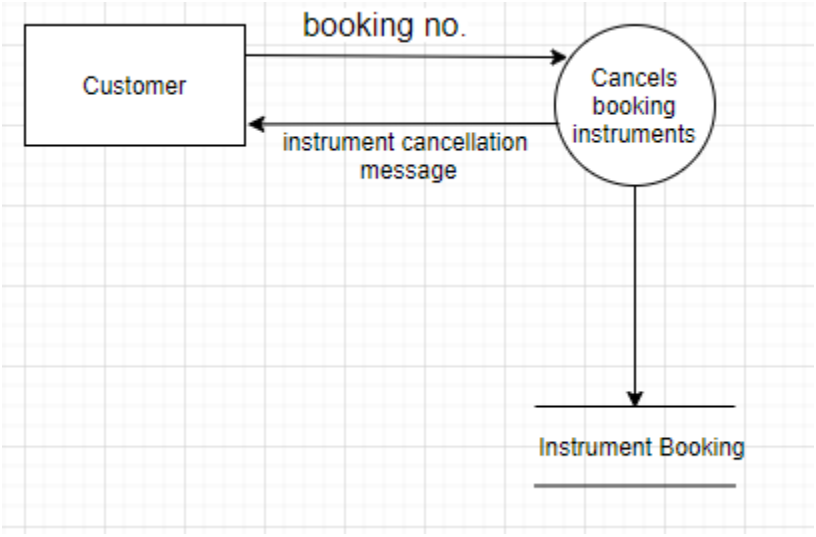


Figure 22: Customer Cancels Instruments Booking

- DFD fragments for “Member gets special discount.”

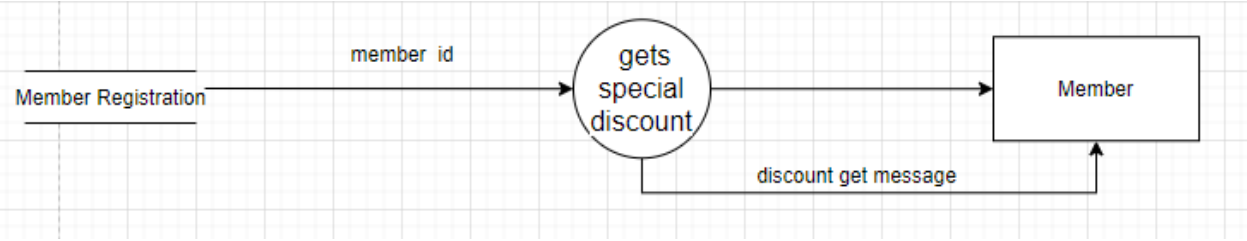


Figure 23: Member Gets Special Discount

3.2 Internal Model Specification

3.2.1 DFD Level 1

Level 1 DFD is simply an exploded view of the context level diagram. It is clear that a context level diagram represents the whole system in a single main process. Level 1 DFD breaks down the main process into subprocesses that can then be seen on a deeper level. Also, level 1 DFD contains data stores that are used by the main process to store an amount of data which is required by the system for further usage (Valcheva, 2019).

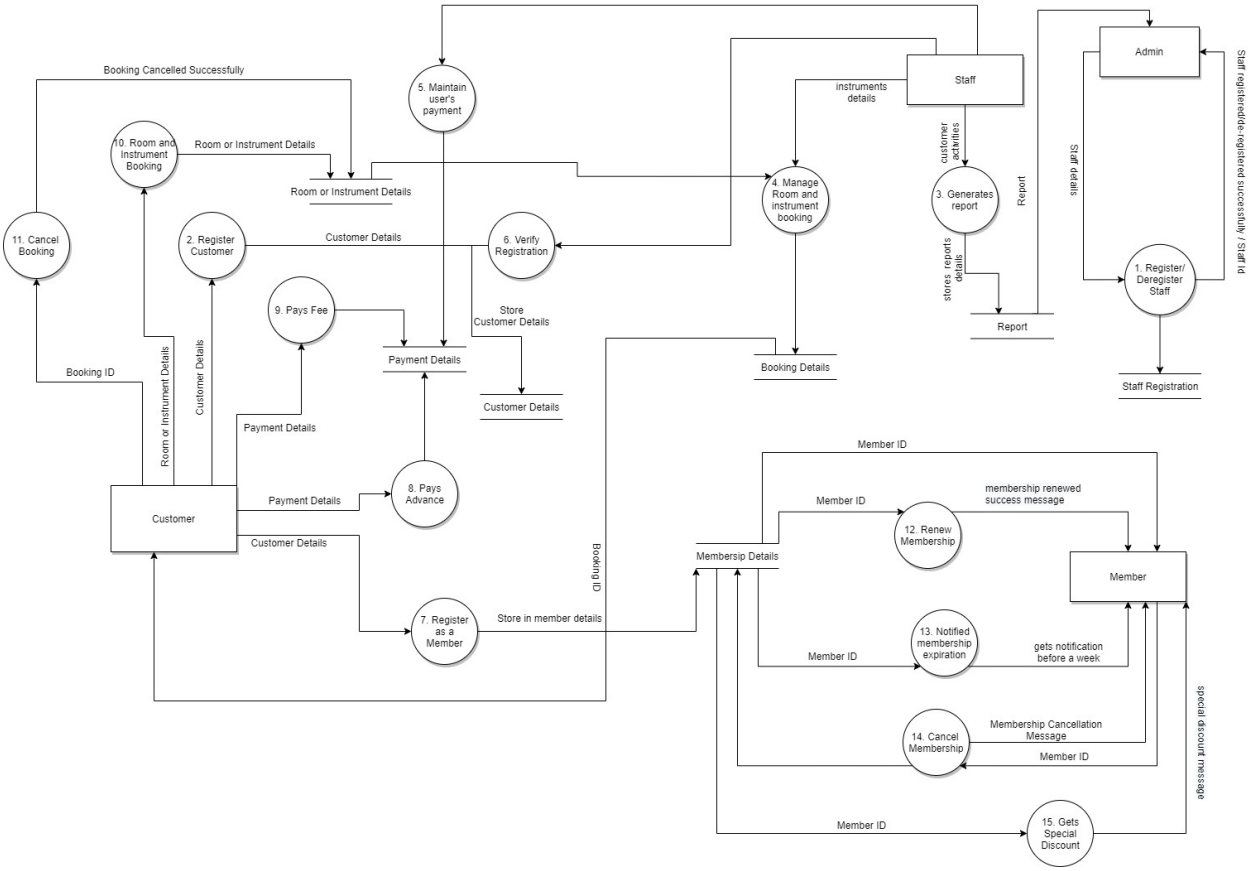


Figure 24: DFD Level 1

Above shown diagram is the Level 1 DFD of a Sound Strong Application system. Firstly, a context level diagram of the system was created and analyzed properly. Further, the main process of the system was fragmented to many subprocesses by which the Level 1 DFD was created. Each process in the Level 1 DFD must have its own appropriate data flow and various data storages are also being used. All of the processes, external

entities, data storages and data flow are well labelled in the mentioned figure and also the data flow of a system is clearly explained.

3.2.2 DFD Level 2

DFD level 2 goes one step deeper into parts of DFD level 1. It can be used to plan or record the specific/necessary detail about the system's functioning (GeeksforGeeks, 2020). The DFD level 2 of the Sound Strong System is shown below,

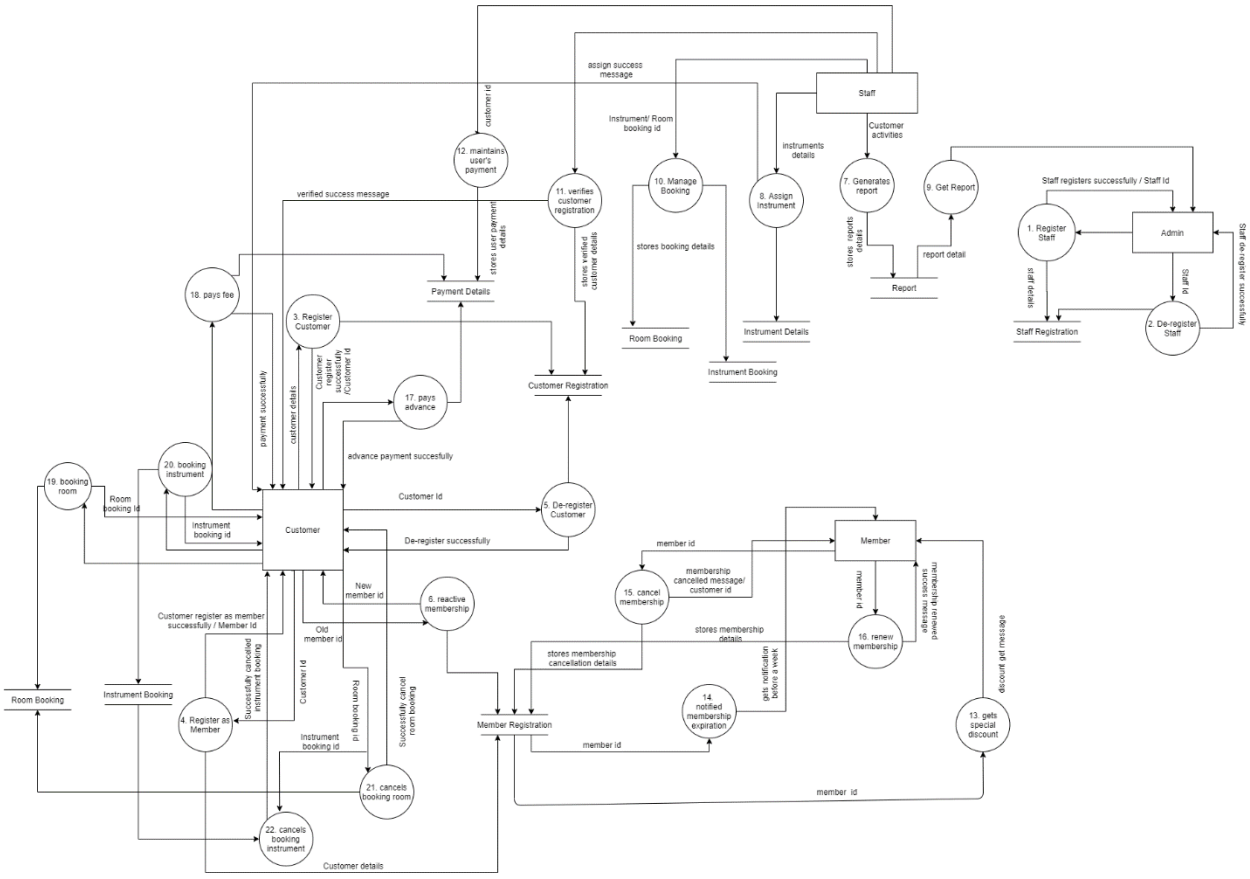


Figure 25: DFD Level 2

Context level diagram and DFD level 1 made earlier was not explained the process of the system properly. So, DFD level 2 was made to explain properly about the process of the system. The entities, process and data store used in DFD was explained well and labelled properly in DFD level 2. The data flow in the Sound Strong Application was properly shown and clearly described in DFD level 2. The data flow of the system is as below:

- Admin register and de-register staff in the system.

- The details of the customer are provided to the system i.e. Sound Strong Application and then Staff verify the registration and generates report according to the customer activities and sends the report to the admin.
- Customer can de-register from the system.
- Customer can buy membership and can register them as member in the system.
- The time of the membership is 3 months and system notify to the member about their membership expire before a week.
- The staff assign new instruments arrived in the store in the system.
- The staff manages the booking made by the customer like booking cancellation.
- Customer can book instruments and rooms as per their demand.
- Customer can cancel instruments and room booking.
- Customer can reactive their expire membership.
- Customer should pay advance before booking rooms and instruments.
- Member can cancel their membership and can renew their membership.
- Member gets special discount for rooms and instruments booking.

3.2.3 Entity Relationship Diagram (ERD)

An entity relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between entities within the system. It is used in software engineering during the planning stages of the software project. It helps to identify different system elements and their relationship with each other (Nishadha, 2020). ERDs are most often used to design or debug relational databases in the fields of software engineering, information systems, business etc. ER Diagrams are composed of entities, relationships and attributes (Lucidchart, 2020). It contains different symbol, diagram and connector that visualize two important information major entities and the inter-relationship among these entities. The ERD of Sound Strong System is shown below,

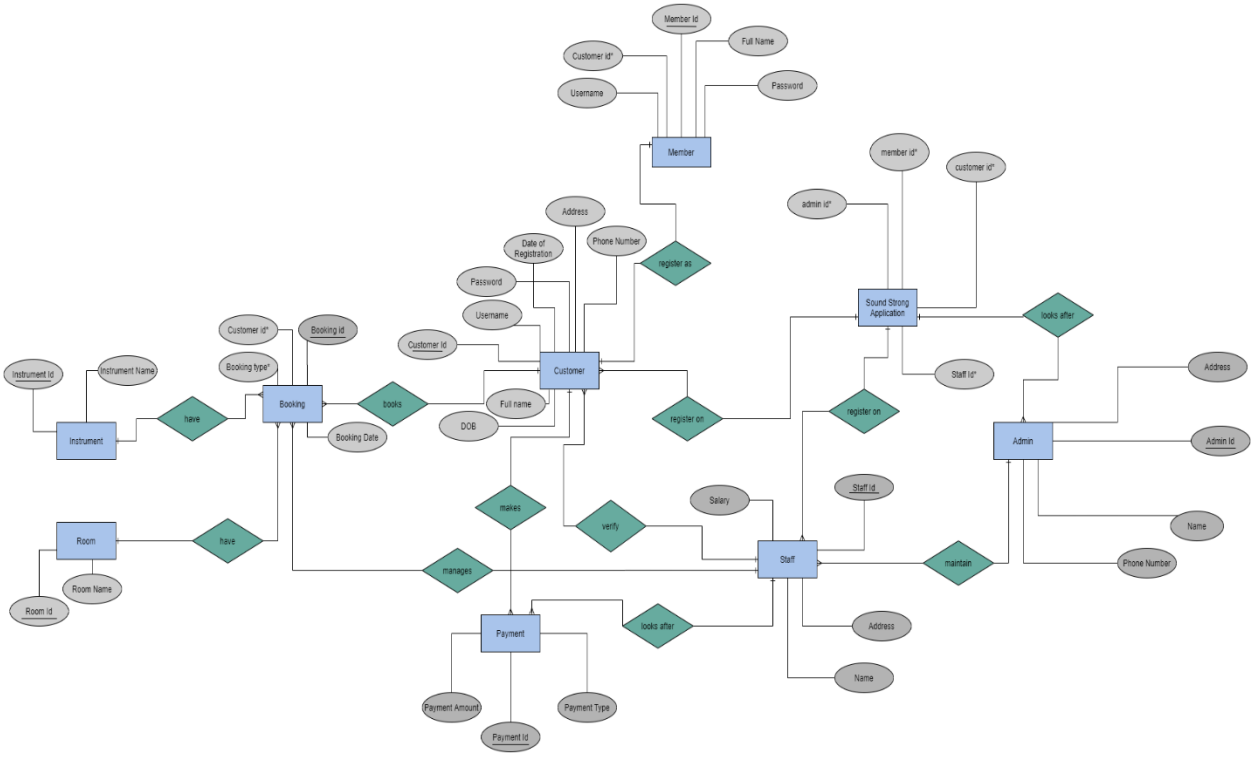


Figure 26: ERD of Sound Strong System

There are seven entities in the Sound Strong Application i.e. sound strong application, admin, member, staff, customer, payment and booking. Each entity has their own attributes. The relationship between entities is shown in above ERD. The relationship between customer and member is one to one, customer and booking is one to many, customer and payment is one to many, customer and staff is many to one, customer and sound strong application is many to one, sound strong application and staff is one to many, sound strong application and admin is one to many, admin and staff is one to many, staff and booking is one to many, instrument and booking is one to many, and room and booking is one to many.

3.2.4 Data Dictionary

A database dictionary provides metadata, i.e. database info. The dictionary is very important because it includes information such as what is in the database, who can access it, the physically stored database etc. Database users don't usually communicate with the data dictionary. The database managers just deal with this.

- General information on the following is in the data dictionary.
- Names and patterns of all database tables.
- Details of all database tables, such as its owners, their security vulnerabilities, when they were developed, etc.
- Physical details on tables like the location and how.
- Table restrictions, including key attributes, knowledge of international key etc.
- Database views knowledge that is available (Meador, 2018).

Sound Strong Application

Column	Data Type	Description	Example
Admin Id	varchar (10)	Primary key of Admin table acts as a reference key	AD001
Customer Id	varchar (10)	Primary key of Customer table acts as a reference key	CUS001
Staff Id	varchar (10)	Primary key of Staff table acts as a reference key	ST001
Member Id	varchar (10)	Primary key of Member table acts as a reference key	MB001

Table 4: Data Dictionary of Sound Strong Application

Admin

Column	Data Type	Description	Example
Admin Id	varchar (10)	Primary key of Admin table	AD001
Name	varchar (20)	Full name of Admin	Nishan Kadel
Phone number	number	Phone number of admin	9816761692
Address	varchar (20)	Address of admin	Itahari

Table 5: Data Dictionary of Admin

Staff

Column	Data Type	Description	Example
Staff Id	varchar (10)	Primary key of Staff table	ST001
Name	varchar (20)	Full name of Staff	Bibek Pradhan
Address	varchar (20)	Address of staff	Jhumka
Salary	number	Salary of staff	90000

Table 6: Data Dictionary of Staff

Payment

Column	Data Type	Description	Example
Payment Id	varchar (10)	Primary key of Payment table	PM001
Payment Amount	number	The amount paid by the customer to Sound Strong	50000
Payment Type	varchar (20)	Type of the payment done by the customer (e.g. debit card, credit card, cash)	Esewa, Visa Card, Khalti

Table 7: Data Dictionary of Payment

Booking

Column	Data Type	Description	Example
Booking Id	varchar (10)	Primary key of Booking table	BK001
Customer Id	varchar (10)	Primary key of Customer table acts as a reference key	CUS001
Booking Type	varchar (20)	Primary key of instrument and room table acts as a reference key	RO001, INS001
Booking Date	Date	Date of booked items	12-October-2020

Table 8 :Data Dictionary of Booking

Customer

Column	Data Type	Description	Example
Customer Id	varchar (10)	Primary key of Customer table	CUS001
Full name	varchar (20)	Full name of customers	Sachin Dahal
DOB	Date	Date of birth of customer	12-July-2000
Username	varchar (20)	Username of customer	iamsachindahal
Password	varchar (30)	Password of the username	**********
Date of Registration	Date	Registration date of customer to system	16-December-2019
Address	varchar (20)	address of customer	Dharan
Phone Number	number	Phone number of customers	98012345678

Table 9: Data Dictionary of Customer

Member

Column	Data Type	Description	Example
Member Id	varchar (10)	Primary key of Member table	CUS001
Full name	varchar (20)	Full name of members	Ashes Rai
Username	varchar (20)	Username of members	ashraiofficial
Password	varchar (30)	Password of the members	**********
Customer Id	varchar (10)	Primary key of Customer table acts as a reference key	CUS001

Table 10: Data Dictionary of Member

Room

Column	Data Type	Description	Example
Room Id	varchar (10)	Primary key of room table	RO001
Room name	varchar (20)	Full name of room	Pablo, Gustavo, Escobar

Table 11: Data Dictionary of Room

Instrument

Column	Data Type	Description	Example
Instrument Id	varchar (10)	Primary key of instrument table	INSS001
Instrument Name	varchar (20)	Full name of instrument	Guitar, Piano

Table 12: Data Dictionary of Instrument

3.2.5 Process Specification

The process specification is a tool for recording, evaluating and describing the logic for making decisions and formulas for producing output data from input process data. Its goal is to flow across and define specifications and process for regulatory / engineering. Clear and full process requirements would be required for high quality, reliable data. (techopediA, 2020)

Register a customer

Number: 1

Name: To register a customer

Description: To register the details of a customer by the Staff

Input data flow: Requires Customer details

Output data flow: Stores the details of a registered customer

Process: Online

Process logic:

Insert registered customer details

Verify the details

Store data.

Register Staff

Number: 2

Name: Register Staff

Description: To register staff

Input flow data: Handles the registration of a staff

Output flow data: Verifies registration

Process: Online

Process logic:

Admin access required

Insert Staff's details

Store registration

De-register Member/Customer

Number: 3

Name: De-register member/customer

Description: To de-register member/customer from the system

Input flow data: Handles the deregistration of a member/customer

Output flow data: Verifies deregistration

Process: Online

Process logic:

User accesses their account

Initiates their deactivation process

Verifies deregistration

Delete data of the respective user.

De-register staff- deregistering the staff that are no longer a staff.

Number: 4

Name: De-register staff

Description: To de-register a staff which are no longer a staff of Sound Strong.

Input data flow: Requires staff's unique id

Output data flow: Verifies deregistration of the staff

Process: Online

Process Logic:

Accessing Staff deregistration by Admin

Admin inputs the staff unique id

If the staff exists deletion of the de-registered staff's data is initiate or else error message is shown

Verifies deregistration

Book the practice room

Number: 3

Name: Add Booking

Description: To maintain the payment of a customer

Input flow data: id for booking and user

Output flow data: Books the room for customer

Process: Online

Process logic:

Id for booking is inputted into the function

Availability is checked in both instrument and room data stores and if it not available all the other options are showed

If it is available, then id is checked to find whether it is room/instrument and if it is room then membership is checked or else if it is not available.

if user is a member then a discount is offered and two options of payment online payment and cash on Arrival else user is checked to see if he has booked more than 8 times or not, if user has then they will be given the same option that a member gets minus the discount or else if user is not a member and also if user has not booked more than 8 times, he will have to pay at least some advance fee.

All payment data is stored afterwards in the payment details, and room booking data stores.

Display the user's payment details and his bookings.

Book preferred instruments

Number: 3

Name: Add Booking

Description: To maintain the payment of a customer

Input flow data: id for booking and user

Output flow data: Books the instrument for customer

Process: Online

Process logic:

Id for booking is inputted into the function

Availability is checked in both instrument and room data stores and if it not available all the other options are showed

If it is available, then id is checked to find whether it is room/instrument and if it is instrument then membership is checked or else if it is not available.

if user is a member then a discount is offered and two options of payment ONLINE PAYMENT and cash on Arrival else user is checked to see if he has booked more than 8 times or not, if user has then they will be given the same option that a member gets minus the discount or else if user is not a member and also if user has not booked more than 8 times, he will have to pay at least some advance fee.

All payment data is stored afterwards in the payment details, and instrument booking.

Display the user's payment details and his bookings

.

Maintain user/payment detail

Number: 4

Name: Maintain user/payment detail

Description: Maintains all users details and their payment details.

Input data flow: Requires customer id

Output data flow: Shows all the data of the user

Process: Online

Process Logic:

Input of customer id

Get all the data of user from customer registration and payment details

Display all the information, if the customer id does not match it shows error.

Generate report

Number: 4

Name: Generate report

Description: Staff generates report

Input data flow: Customer activities

Output data flow: Generates report

Process: Online

Process Logic:

Input taken from staff

Generate report by using the input from staff as reference

Store the report in report data store

Notify customers

Number: 4

Name: Notification controller

Description: Notifies the customer of their membership expiration before a week and their reserved time half an hour before.

Input data flow: Null

Output data flow: Generates push notification

Process: Online

Process Logic:

Notification controller module is run every 60 seconds in a loop

Checks whether membership expiration date is in a week from current time and date if expiration date is in a week it sends notification to the user else nothing happens.

Checks whether reservation time is in half an hour from current time and if reserved time is in half an hour it sends notification to the user else nothing happens.

Manage bookings

Number: 4

Name: Manage bookings

Description: Staff can view and manage reserved rooms and instruments time allocations

Input data flow:

Output data flow: Generates successful message with altered data

Process: Online

Process Logic:

Staff can choose either to view or manage instrument or room booking

If staff chooses to manage instrument or room booking else display data from instrument or room booking

If timings are reallocated or cancelled, then it is stored in their respective data stores and display successful message with new time allocations

3.3 Design Specification

3.3.1 Structure Chart

Structure chart is hierarchical graphical representation of the decomposition of any given problem, it is mostly used as a tool in software development to facilitate the coding process of a system. Structure chart breakdowns the system in a more manageable sub-levels which are called sub-modules; which essentially helps coders, software engineers, system administrator by making a big system into a more digestible and manageable chunks. (Bell, 2005)

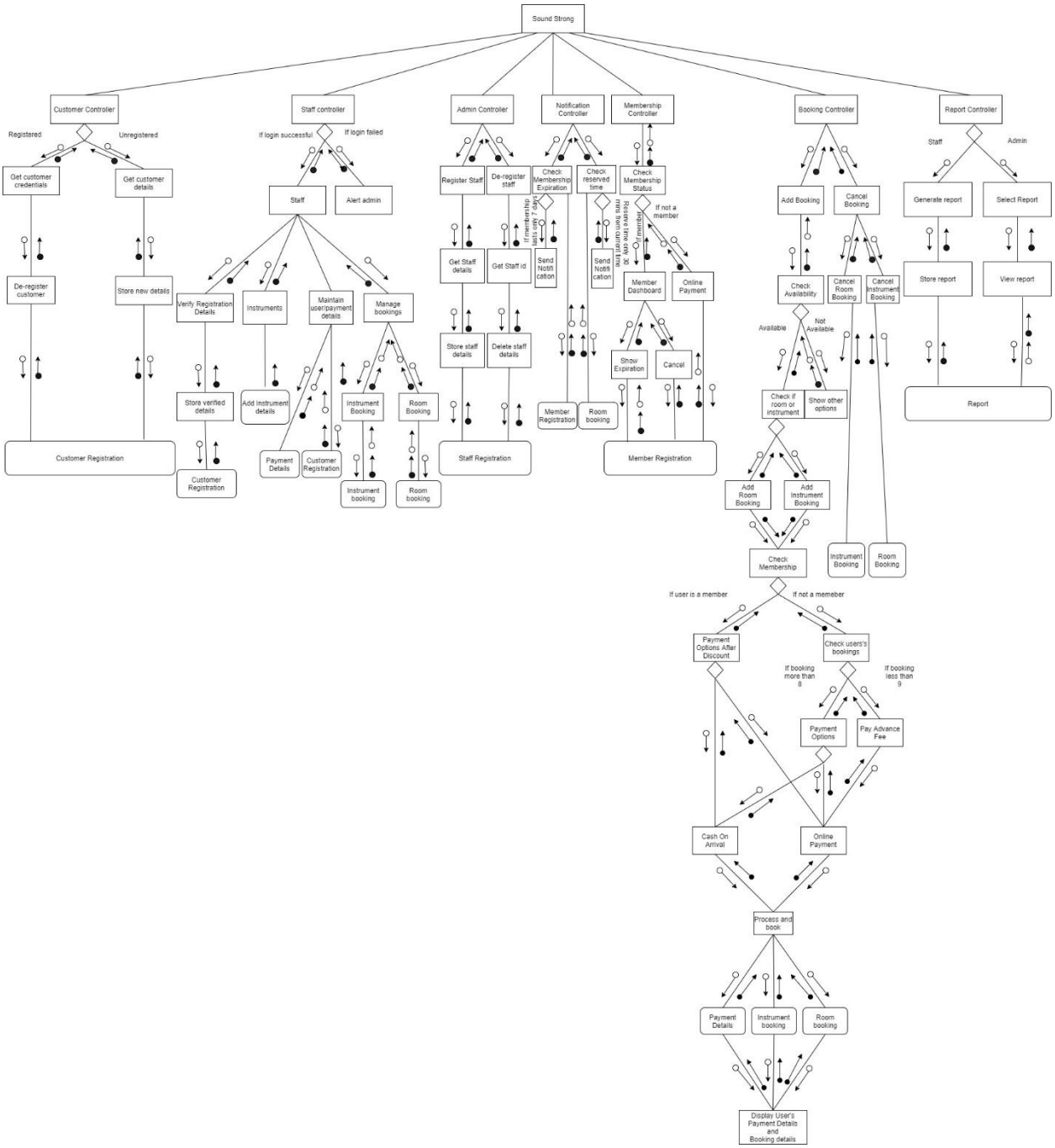


Figure 27: Structure Chart

The above figure is a structure chart for the Sound Strong music company's management system. The chart shows that the system is divided into 7 modules which are then divided into submodules and other submodules. The explanation of the 7

modules of the Sound Strong's management system is listed below with their brief explanations:

1. Customer Controller: Customer controller module logs user into the system and allows an option to de-register or perform other various actions with the integration of other modules. Another feature of customer controller module is that it registers the new user to the system for further verification by staffs.
2. Staff Controller: Staff controller module asks for the login of the staff and if login is not authorized then it reports to the system admin to ensure the safety and security of the system. If the login of staff is authorized by the system then staff has access to verify the new registered users, add new instruments, manage the timings of the bookings, and manage and view user and payment details.
3. Admin Controller: Admin controller module has just two submodules without any if else condition. First submodule is register staff which asks for the insertion of staff's details and registers staff into the system. Second submodule is for deregistering staffs and is called deregister staff, this module takes the staff unique id and deregisters the staff from the system.
4. Notification Controller: Notification controller recursively runs itself every 60 seconds. It at first checks incoming expiration date of members that is within a week and also checks incoming reservation time that is within 30 minutes. If there are members with incoming expiration date in a week a notification is sent to them and if reservation time of any user is within 30 minutes that user is also notified.
5. Membership Controller: This module checks the membership status of the user and if the user is not a member, they can pay with online payment to get a membership of the Sound Strong or if the user has less than 7 days remaining in the membership plan then a notification for the user is generated and send to the appropriate user, else if user is a member, he can see the remaining time on his membership or cancel the subscription which takes in effect after the user's membership ends.
6. Booking Controller: This is the biggest module among all the modules and it essentially handles all the payments and booking for the Sound Strong company. There are two submodules inside booking controller, and they are add booking

and cancel booking. They each have their own submodules which aids them in their doing their tasks. Add booking allows user to book instrument and rooms if it is available for renting or else it shows other options that is available to the user. After booking an available room or instrument users are checked whether they have membership or not if they do not have membership their booking history is checked to confirm that they have booked at least 8 times before current booking if user has booked 8 times prior to current booking they are given the same options as the member for payment which is online payment or cash on arrival after that everything is processed booking is displayed to the user and also added to the database or other forms of storage media. If a user has not booked the room 8 times prior to current booking they have to pay at least some advance fee through online payment to book and only after that they are allowed to book. After which the booking is processed, displayed, and added to the storage media. On the other hand, cancel booking is very straightforward. There are two submodules inside cancel booking which are cancel room booking and cancel instrument booking which allows you to cancel your room booking or instrument booking respectively.

7. Report Controller: This is a relatively small module compared to other modules it has just two submodules. First submodule is generate report and it is only accessible to the staff, staffs then can store the generated report in the system. Second submodule is select report which is only for the admin, select report takes the input from the admin and generates report into a more user-friendly format using view report submodule of select report.

3.4 Assignments Diary

3.4.1 Assumptions

To develop the fully functioning system of Sound Strong Application, few assumptions were made at the beginning as well as implemented in the project for the better performance and consistency of the product. Following are the assumptions made:

- There will be no discount scheme for the normal customers.
- The assumption of the customer username and password is made to login to a system after getting registered.
- A customer will be 50% refunded in cancellation of the booking.
- Customers who deregister themselves from a system can again apply for the registration.
- Membership will last till 3 months and the member will be notified about it before a week of their membership expiration.
- Payment details of a customer will be stored in a system in an encrypted form.

3.4.2 Omissions

Following are the inconsistencies that we found in the project:

- A company staff cannot deregister themselves without the involvement of the admin in this process.
- A customer can register to the system without paying any fees.
- A de-registered customer's detail will be auto deleted after a week of getting de-registered.
- A customer with membership cannot cancel the membership till their 3 months' time got fulfilled.

3.4.3 Meeting Log

To complete this groupwork we five group members conduct many meetings and discusses about the problems. The meeting logs are given below:

Meeting Log 1

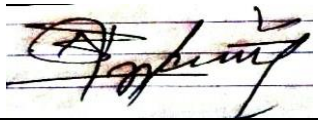
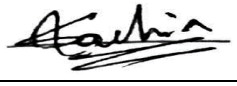
Meeting No: 1 **Date**: 16-12-2020

Start Time: 3:30 PM **Finish Time**: 4:30 PM

Items Discussed: General idea of a given groupwork

Objectives: As this was the first meeting of us, we discuss about the general idea of a given group work, how to divide the task and proposed a kind of our working roadmap.

Tasks for Next Meeting: To divide the task of the group work.

Attendees Name	Signature
Nishan Kadel	
Bibek Pradhan	
Ashesh Rai	
Ashmina Rai	
Sachin Dahal	

Meeting Log 2

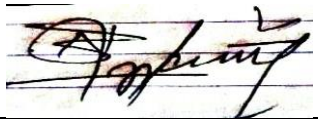
Meeting No: 2 **Date**: 20-12-2020

Start Time: 4 PM **Finish Time**: 5 PM

Items Discussed: Dividing Task to each member

Objectives: Each member of the group was designated with a specific task which is to be completed within the next meeting. Even rough design works for the first time.

Tasks for Next Meeting: Discussion on whether the assigned task has been completed or not.

Attendees Name	Signature
Nishan Kadel	
Bibek Pradhan	
Ashesh Rai	
Ashmina Rai	
Sachin Dahal	

Meeting Log 3

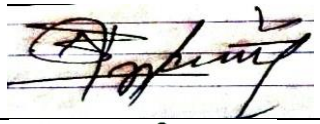
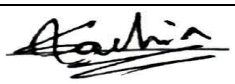
Meeting No: 3 **Date**: 23-12-2020

Start Time: 4 PM **Finish Time**: 4:30 PM

Items Discussed: Discussion in Task Completion

Objectives: We checked each other's task individually and found out all the things were perfectly correct. Still a few things were there to be changed but those were minor and easy to solve ones.

Tasks for Next Meeting: Checking our task and verifying it.

Attendees Name	Signature
Nishan Kadel	
Bibek Pradhan	
Ashesh Rai	
Ashmina Rai	
Sachin Dahal	

Meeting Log 4

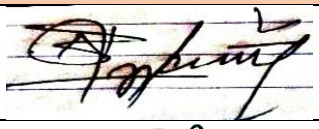
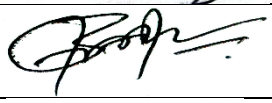
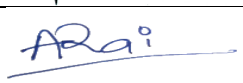
Meeting No: 4 **Date**: 25-12-2020

Start Time: 5 PM **Finish Time**: 6 PM

Items Discussed: Verifying the task

Objectives: In the 4th meeting of us, we verified our completed task by revising it and checking again and filled with the required things which were missed at the first attempt. Everyone explained the concepts of their task and how did they manage to do it.

Tasks for Next Meeting: Finalizing the main documentation file.

Attendees Name	Signature
Nishan Kadel	
Bibek Pradhan	
Ashesh Rai	
Ashmina Rai	
Sachin Dahal	

Meeting Log 5

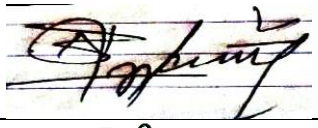
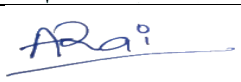
Meeting No: 5 **Date**: 27-12-2020

Start Time: 8:30 PM **Finish Time**: 10 PM

Items Discussed: Finalizing the documentation

Objectives: In the official final meeting of ours, we first set up the format and style sheet for our documentation purpose and finalize the document. We everyone perfectly set the documentation by putting acknowledgement, abstract, group work and individual task.

Tasks for Next Meeting: On the coursework submission date.

Attendees Name	Signature
Nishan Kadel	
Bibek Pradhan	
Ashesh Rai	
Ashmina Rai	
Sachin Dahal	

4. Individual Task

To understand each process executed by the programs in simple way, we have decomposed our system to small parts. Those small parts of the program provide deep knowledge of its functionality. The small parts of this program contain context diagram, level 1 DFD, level 2 DFD, structure chart and module specifications.

4.1 Register a Customer

Name: Bibek Pradhan

London Met ID: 19031617

Context Level Diagram

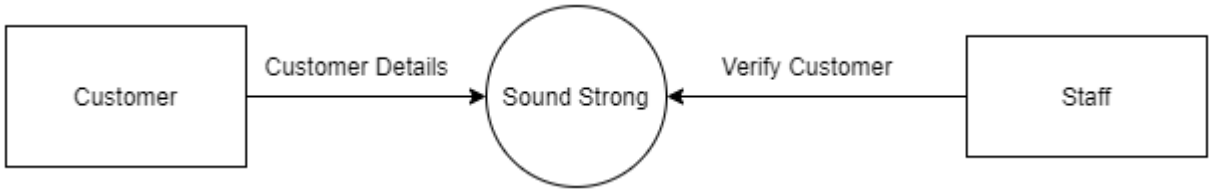


Figure 28: Context Level Diagram of Register a Customer

The above figure is the context level diagram for registering a customer in the system. In this, diagram customer provides their required details to the system and staff verifies the customer registration. After verifying by the staff, customer successfully registered in the system.

DFD Level 1

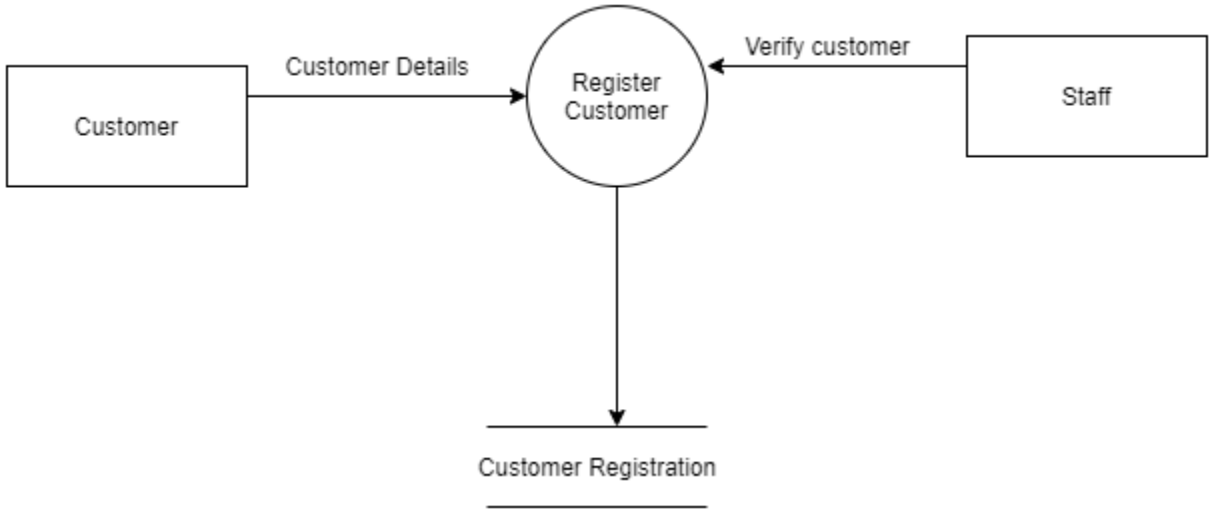


Figure 29: DFD Level 1 of Register a Customer

The above diagram is of DFD Level 1 for registration of the customer. While registration of customer the staff verifies the details of customer and store the registered customer details in the customer registration database.

DFD Level 2

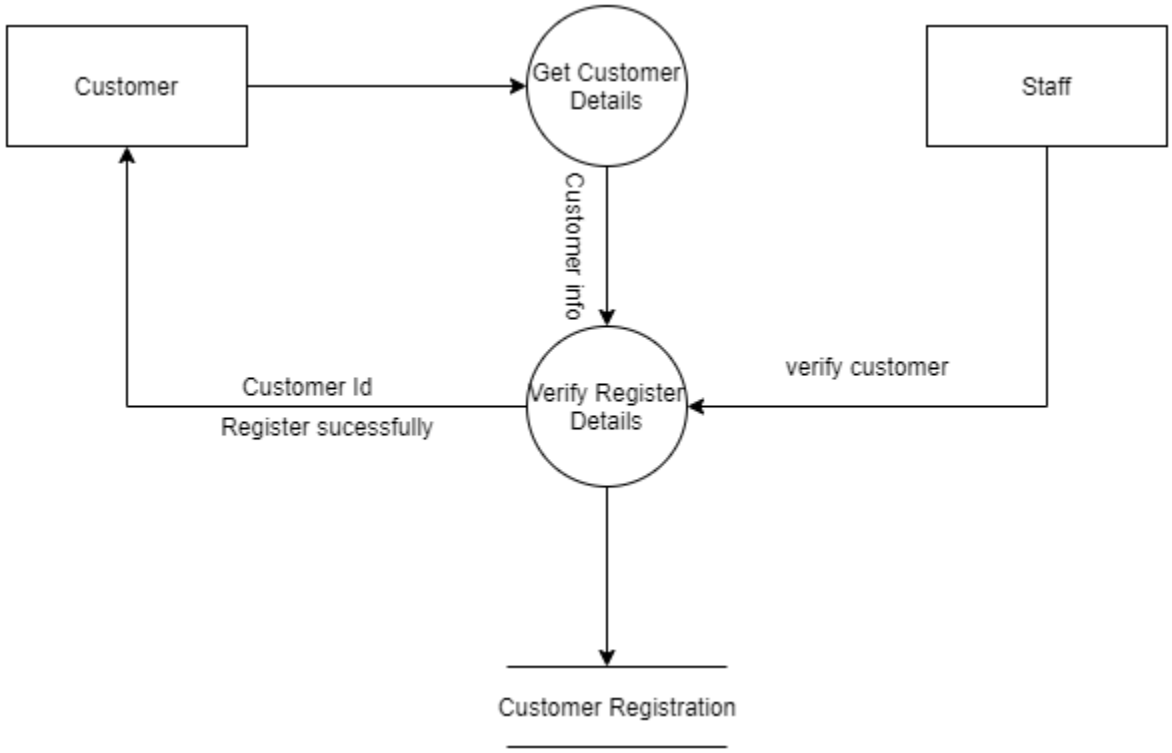


Figure 30: DFD Level 2 of Register a Customer

The above figure is of DFD Level 2 for registration of the customer. At first, customer provides their details to the system and the staff verifies the details of the customer. After verifying by the staff, the details of the customer is store in customer registration database and customer id and register successfully message is send to the customer.

Structure Chart

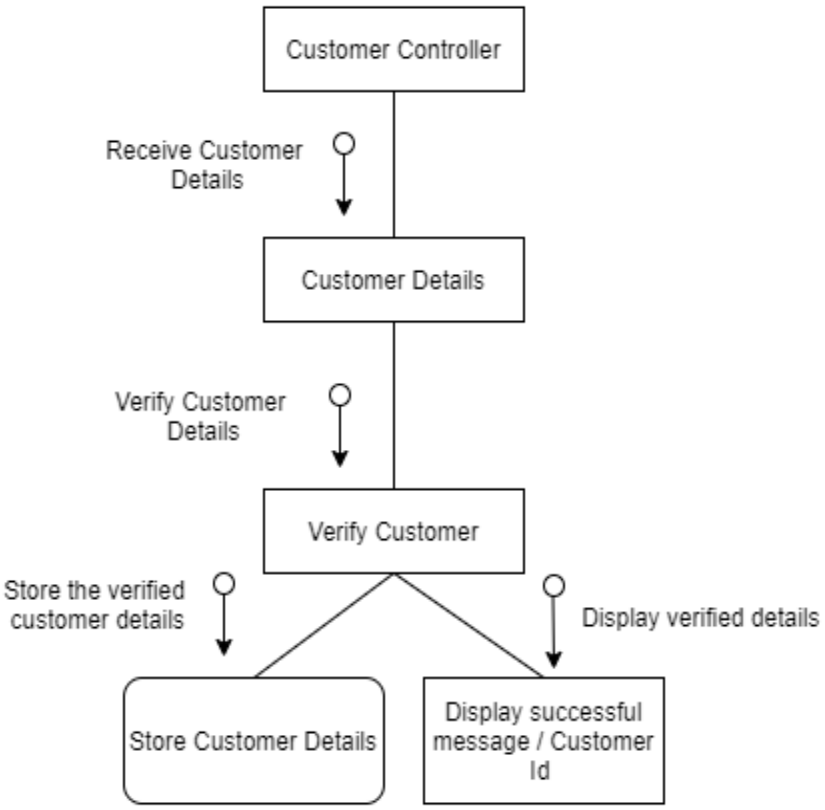


Figure 31: Structure Chart for Register a Customer

The customer controller receives the details of the customer and verifies the customer details. After verifying customer details, verified customer details is stored in database and display verified details.

Module Specification

MODULE NAME: Registration of Customer

PURPOSE: This function is to register the customer in the system.

PSEUDOCODE:

FUNCTION registrationCustomer

GET details of the customer

IF Customer details is verified

Store the verified customer details

Generate random unique number and assign it to the customer as Customer_Id

PRINT “Customer successfully registered.”

ELSE

Reject for further verification

END IF

END

INPUT PARAMETERS: Full_Name, Address, Date_Of_Birth, Phone_Number, Username, Password

OUTPUT PARAMETERS: Username, Password, Customer_Id

GLOBAL VARIABLES: none

LOCAL VARIABLES: none

CALLS: none

CALLED BY: Customer, Staff

4.2 De-register Customer

Name: Nishan Kadel

London Met ID: 19031776

Context Level Diagram

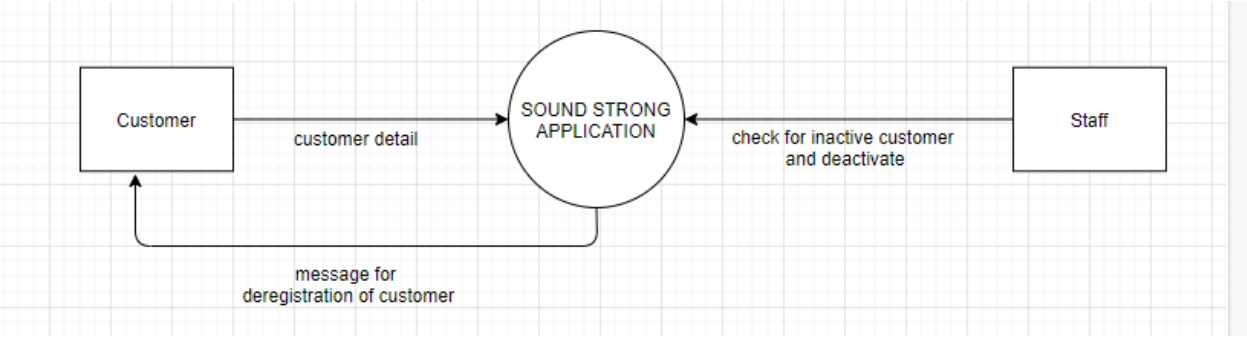


Figure 32: Context Level Diagram of De-Register Customer

The above diagram is a context level diagram for de-registration of customers. With the customer details in the sound strong application, the staff deregister the inactive customer.

DFD Level 1

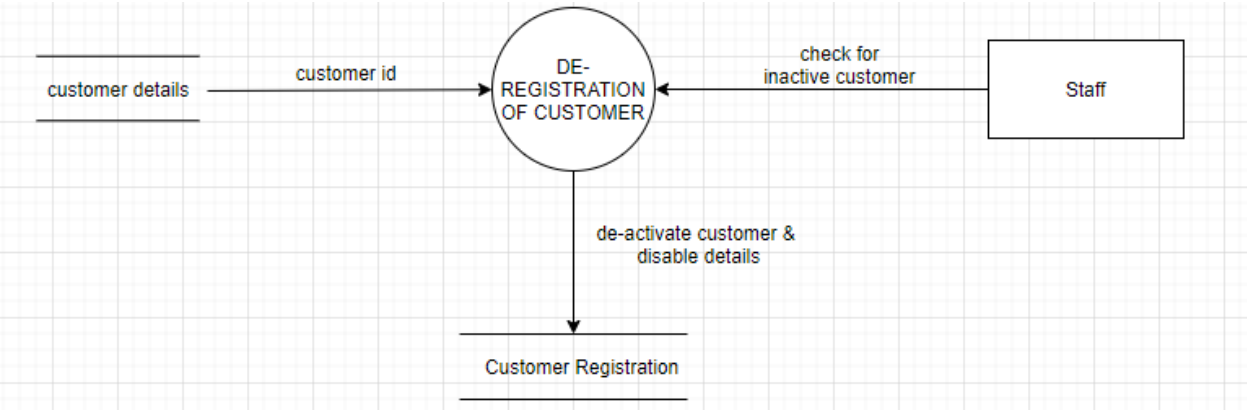


Figure 33: DFD Level 1 of De-register Customer

The above diagram is level 1 DFD for de-registration of customers. In this process the staff access the customer details from the storage and check the status of the customer. If the customer is inactive for a long time, then the staff deregister the respective customer from the sound strong application.

DFD Level 2

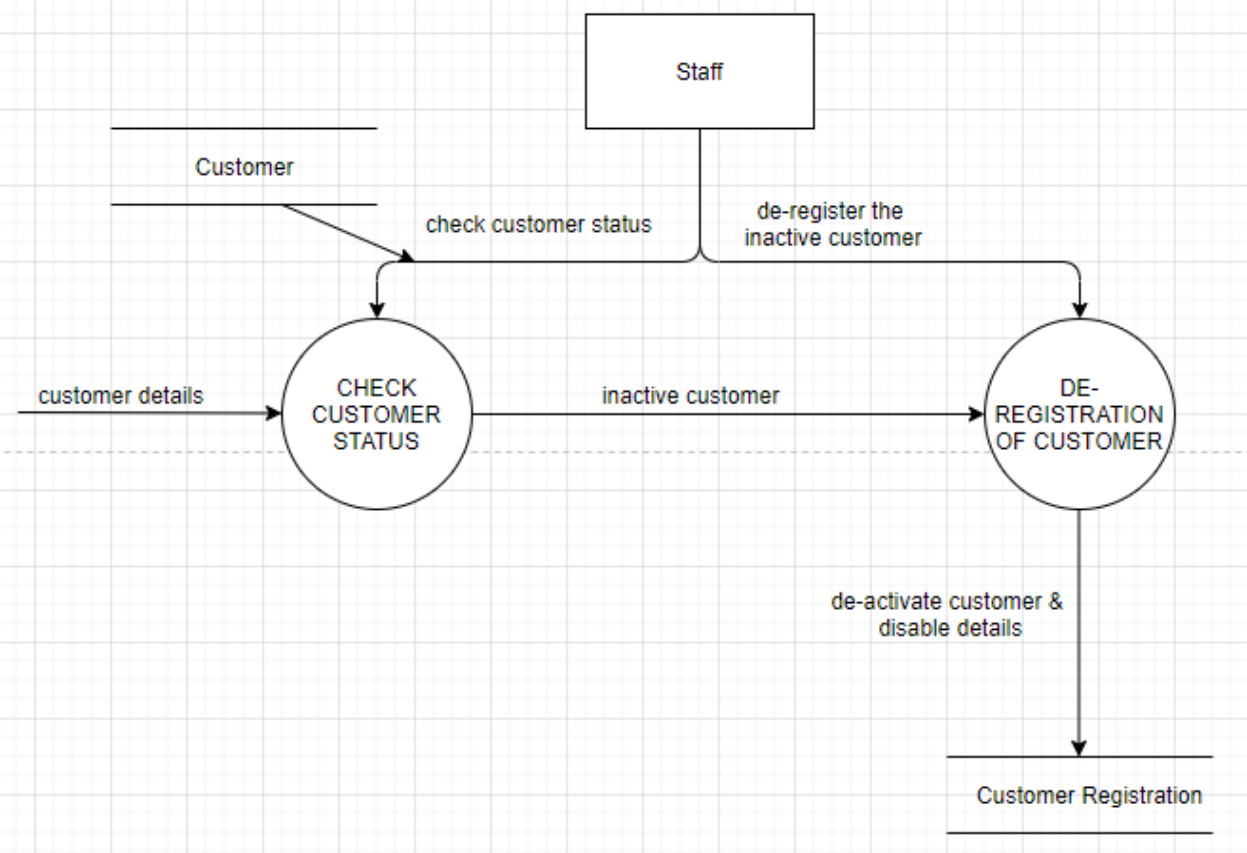


Figure 34: DFD Level 2 of De-register Customer

The level 2 DFD for de-registration of customers is mentioned above. In the above process, the staff accesses the customer details and checks the customer status. If the staff finds that the customer is absent for a long time, then the staff automatically deregisters the customer. After the de-registration, the de-registered customers are stored in the deregistered customer storage.

Structure Chart

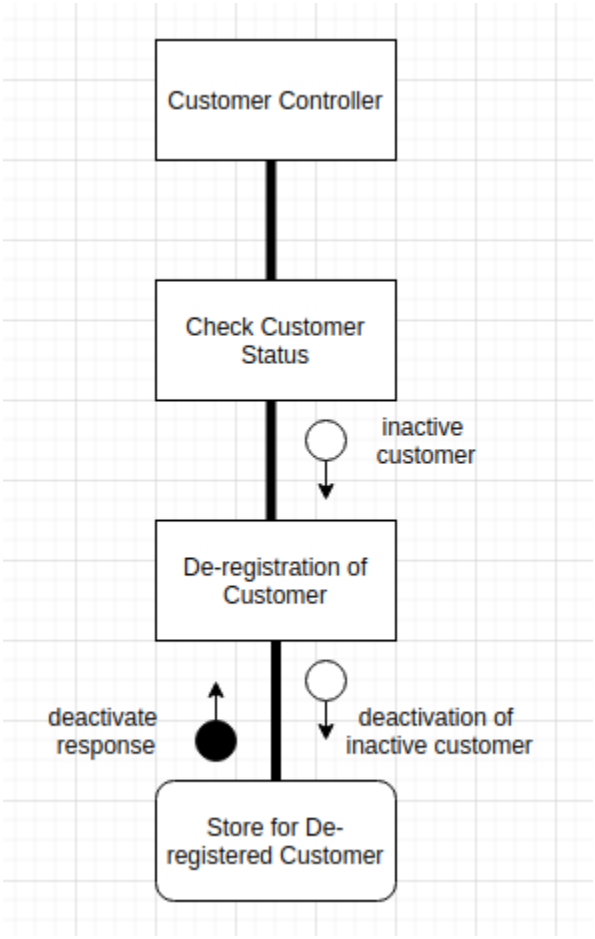


Figure 35: Structure Chart of De-register Customer

First it checks the active status of the customer. If the customer is inactive for a long time or more, then the staff de-registered the customer and stored the de- registered details in a data store as well as gives the response of deactivate details to the system.

Module Specification

MODULE NAME: De-registration of customer

PURPOSE: This function is to deregister the inactive customer

PSEUDOCODE:

The purpose of this module is to de-register a customer from the app.

BEGIN De-register customer

GET customer id, full name, DOB, username, password, date of registration, location

Check the details and status of customer

FUNCTION deregisterCustomer

IF customer is inactive for more a long time

DO De-registration of the specific customer

DO Store of the de-registered customer

DO Delete the customer details

PRINT “Successfully deleted customer”

ELSE IF customer is active

DO nothing

END IF

END

INPUT PARAMETERS: customer id, full name, DOB, username, password, date of registration, location

OUTPUT PARAMETERS: store details of de-registered customer

GLOBAL PARAMETERS: not assigned

LOCAL PARAMETERS: not assigned

CALLS: customer details and staff

CALLED BY: Staff

4.3 Payment of a Customer

Name: Sachin Dahal

London Met ID: 19031839

Context Level Diagram

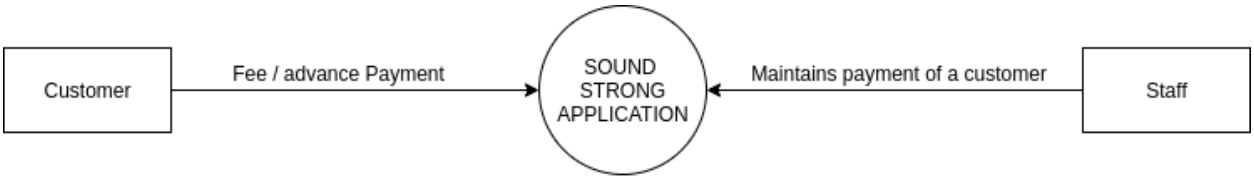


Figure 36: Context Level Diagram of Payment of Customer

Above shown diagram is the context level diagram of the payment of a customer. Here, a customer sends his/her payment details to the system and staff from the system will maintains the payment of a customer.

DFD Level 1

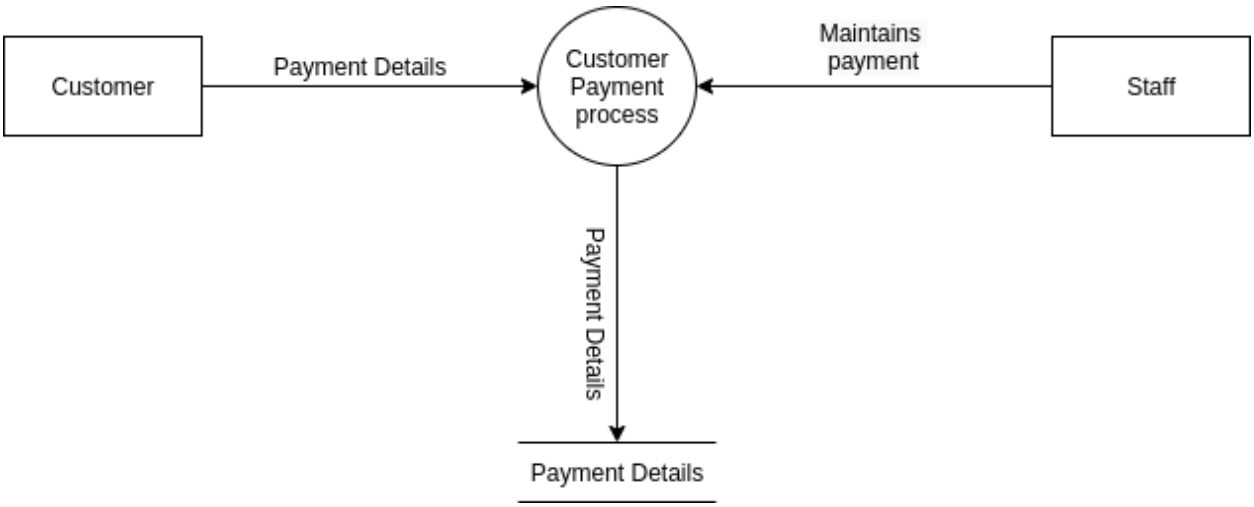


Figure 37: DFD Level 1 of Payment of Customer

Above shown diagram is the Level 1 DFD of the payment of a customer. Here, a customer sends the payment details to the system which goes on the the process of validation first. A company staff will get access to the payment details of a customer and validates the payment details to check whether the payment details entered are correct or not. After the validation of the user's payment, the payment details of a customer are stored in a payment details storage.

DFD Level 2

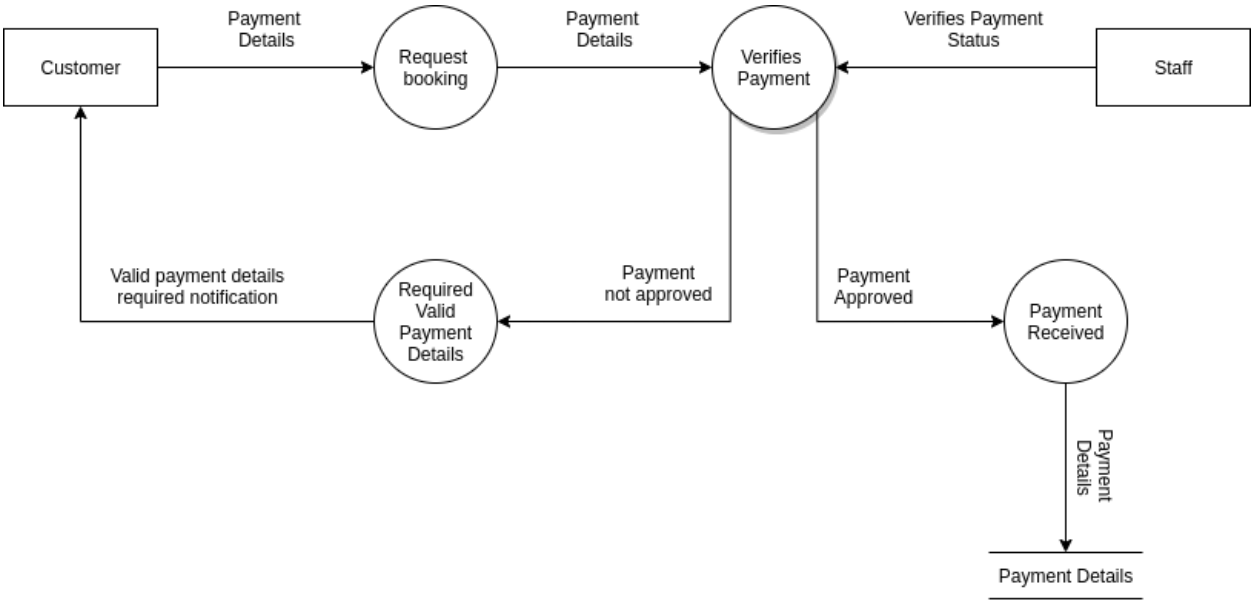


Figure 38: DFD Level 2 of Payment of Customer

Above shown diagram is the Level 2 DFD of the payment of a customer. Here, firstly customer's payment details request for booking is accessed by a company staff and validates it. After validation, if the payment details are approved, company receives the payment and store it in a payment details storage else if the payment details are not approved, customer will be notified with a message to enter the valid payment details.

Structure Chart

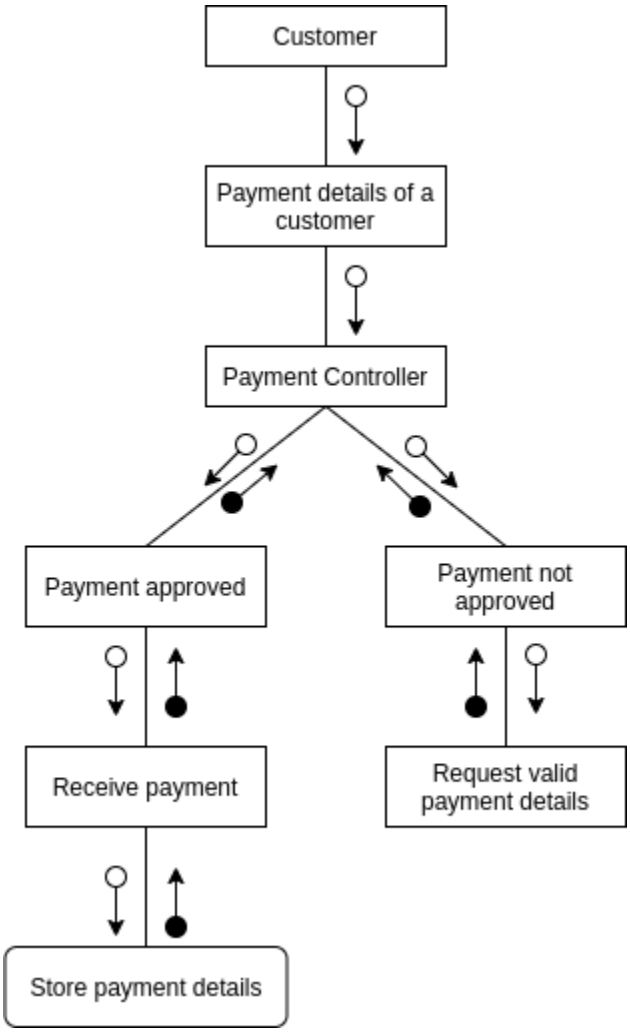


Figure 39: Structure Chart of Payment of Customer

Above shown diagram is the structure chart of the payment of a customer. A customer sends his/her payment details first and then the payment controller validates the payment details entered by the customer. If the details got approved, a company staff will receive the payment from a customer and store the payment details, else if the payment details does not got approved, an appropriate message is send back to the customer requesting them to enter the valid payment details.

Module Specification

MODULE NAME: Payment of a Customer

PURPOSE: A function to check and validate the payment details of a customer

PSEUDOCODE:

```
FUNCTION customer_payment_validation
    GET payment details from a customer
    CHECK payment details
        IF payment details got approved
            Receive payment
            Store payment details
        ELSE
            Request customer to enter valid payment details
        END IF
END
```

INPUT PARAMETERS: Customer Payment Details

OUTPUT PARAMETERS: Payment Details Stored

GLOBAL VARIABLES: Payment Details

LOCAL VARIABLES: none

CALLS: none

CALLED BY: Staff

4.4 Book Practice Room

Name: Ashesh Rai

London Met ID: 19031902

Context Level Diagram

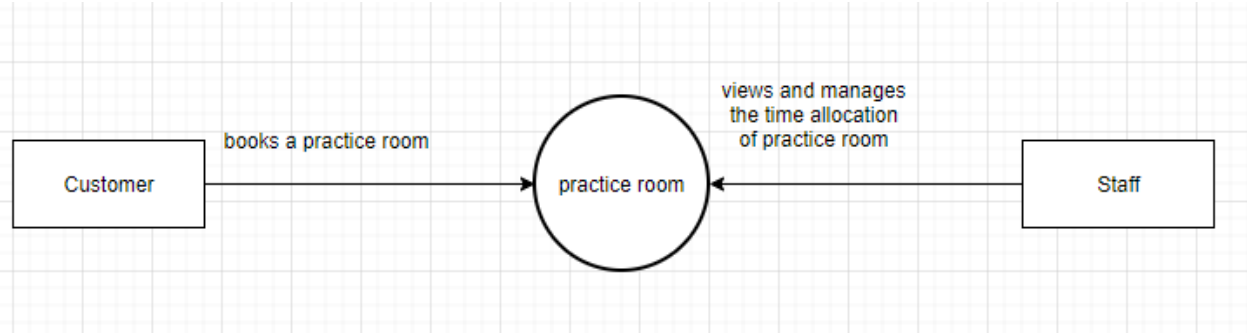


Figure 40: Context Level Diagram of Book Practice Room

In the figure above, context level diagram of book practice room is shown. During the process, a staff views/manages the time allocation for rooms and customer chooses a room from rooms which have free timings.

DFD Level 1

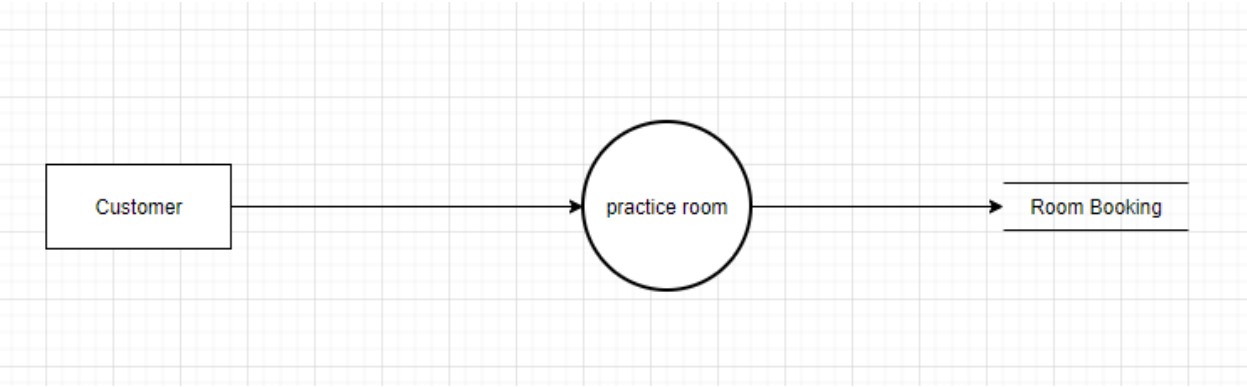


Figure 41: DFD Level 1 of Book Practice Room

Data Flow Diagram Level 1 is shown in the figure above, and it does not have customer entity that is in context level diagram. Customer books the practice room which is then stored in the room booking data store for future references.

DFD Level 2

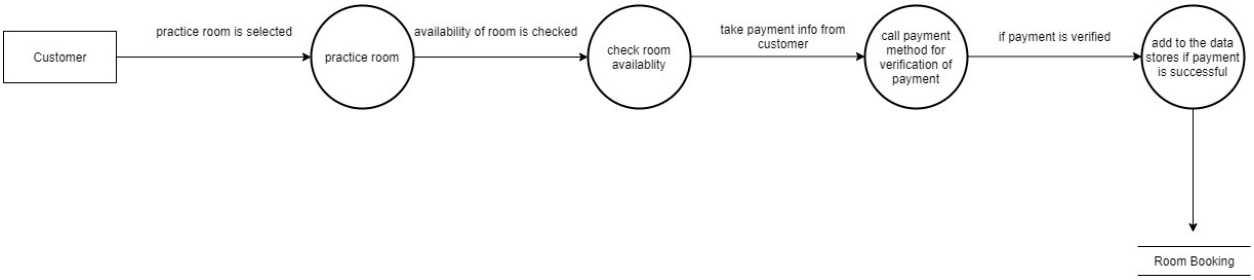


Figure 42: DFD Level 2 of Book Practice Room

In the figure above the Data flow diagram level 2 of Book practice room is shown. A customer chooses a room and its availability is checked by taking data from the room booking data store, if it is available then payment method is called to get payment info from the customer to pay for the booking of the room. If the transaction is verified, then payment info and practice room data is sent to another process that adds booking of the practice room.

Structure Chart

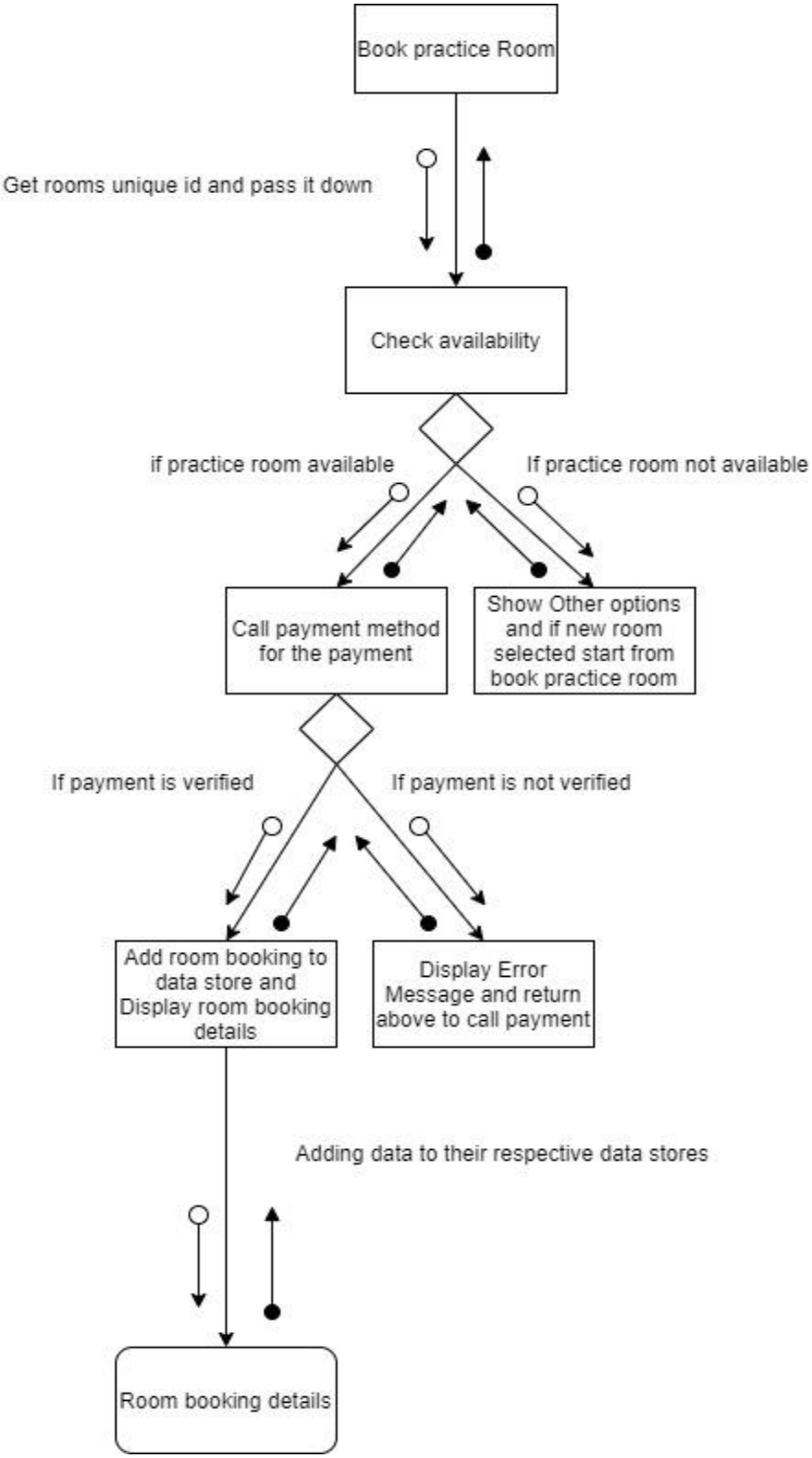


Figure 43: Structure Chart of Book Practice Room

The figure above shows the structure chart of the Book practice room feature. Book practice room shows rooms to all the users and when a user tries to book the room it checks the availability of said room. If the room is available, then payment for the said room is asked by calling the payment method of the system which does all the financial transaction of the system and sends verified messages. If the transaction passes through, then practice room booking is stored in room booking. Also, if the transaction does not pass through then an error message is shown and user is returned back to the payment method.

Module Specification

MODULE NAME: Book practice room

PURPOSE: Browsing and booking practice rooms

PSEUDOCODE:

FUNCTION BookPracticeRoom

SHOW user all the practice rooms

GET user's choice of practice room and user details

IF practice room is available

CALL payment method for payment of the practice room

IF transaction passes through

ADD practice room booking details to the payment details data store.

PRINT "Booking Successful" and booking details, and user details

ELSE

DISPLAY "Booking unsuccessful" or error message

RETURN back to the **CALL** payment method

END IF

ELSE

SHOW all the other rooms

GET the user's practice room choice from other rooms

RETURN to the GET user's choice of practice room and input the room choice taken from above

END IF

END

INPUT PARAMETERS: NULL

OUTPUT PARAMETERS: User details, and Practice room booking details

GLOBAL VARIABLES: None

LOCAL VARIABLES: None

CALLS: Payment by customer

CALLED BY: Customer

4.5 Generate Report

Name: Ashmina Rai

London Met ID: 19031606

Context Level Diagram

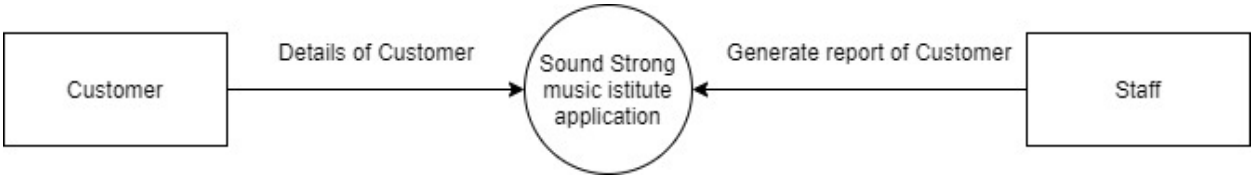


Figure 44: Context Level Diagram of Generate Report

Here, as we can see in the above diagram, a context level diagram is made. During this process, with all the brief details of Customer, staff of sound strong music institute generates the report of a customer.

DFD Level 1

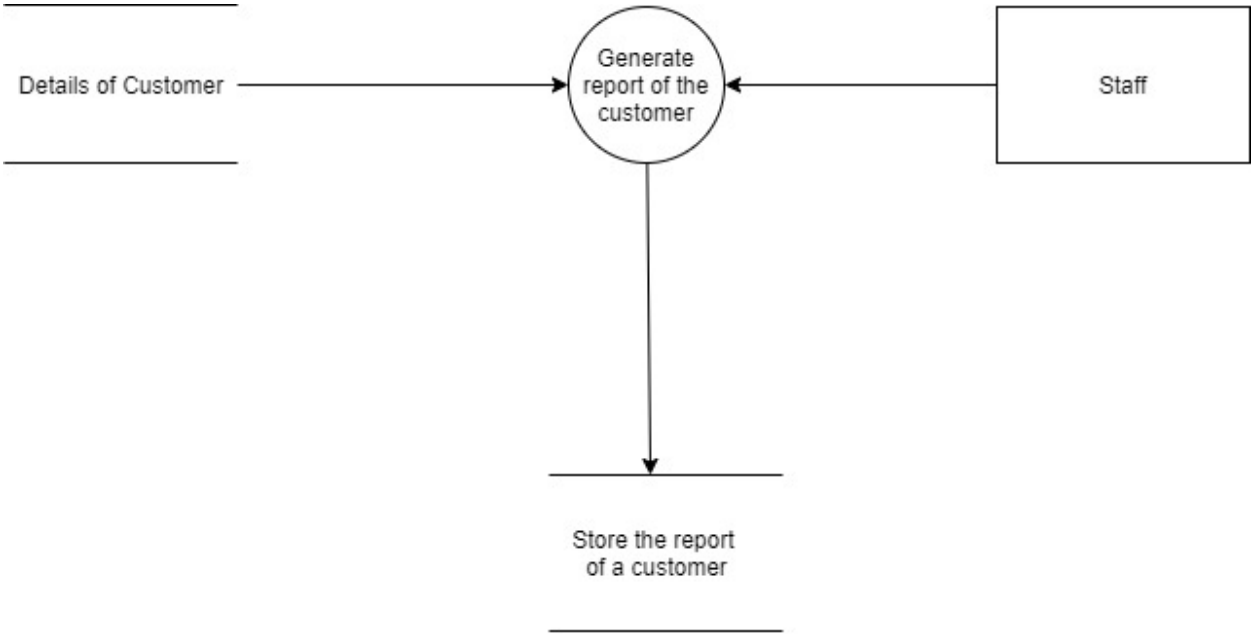


Figure 45: DFD Level 1 of Generate Report

Here, the above picture shows the Diagram of Level 1 Data flow diagram for generate report. During this process, a customer briefly describes about their details from which Staff of sound strong music institute gets the access of a customer and later store the report of a customer.

DFD Level 2

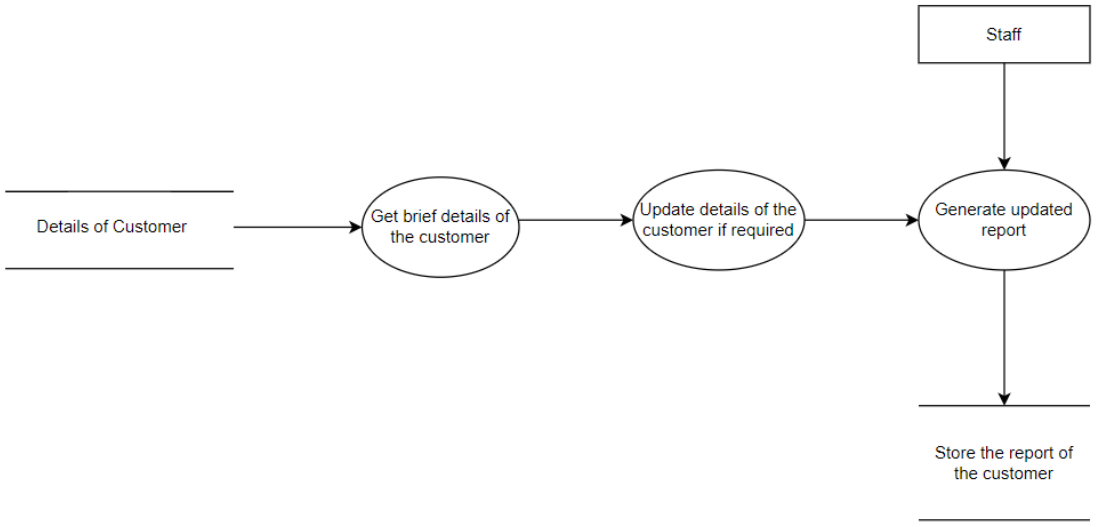


Figure 46: DFD Level 2 of Generate Report

Here, in the above diagram a DFD for level 2 is created. During this process at first details of the customer is taken and updated as well if necessary. Then, a report is generated using both the customer as well as Updated details by the staff of the Sound string music institute and finally a final report is generated and stored.

Structure Chart

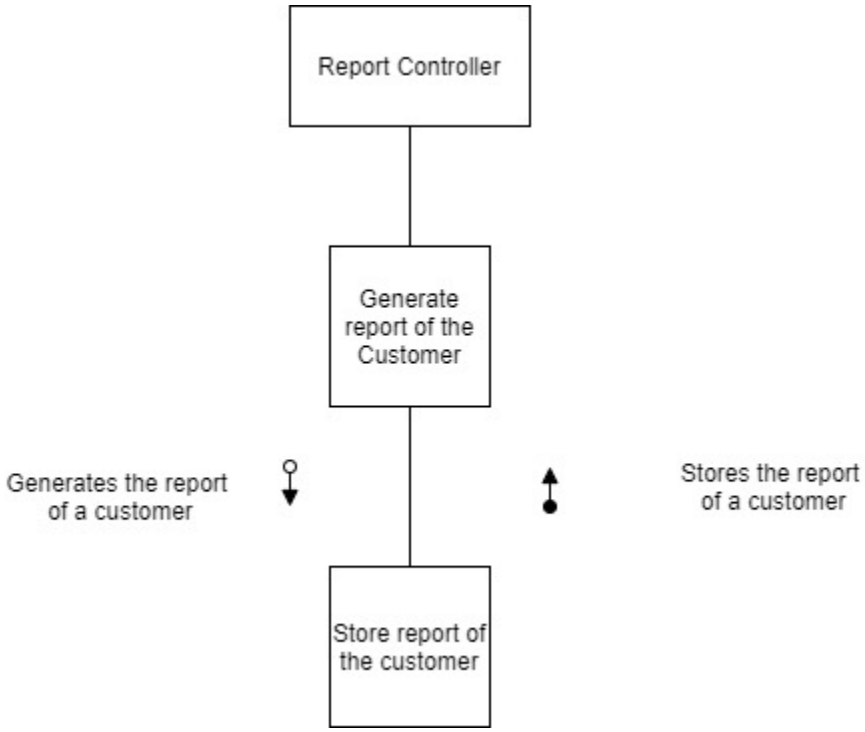


Figure 47: Structure Chart of Generate Report

Here, the above diagram shows the structure chart which generates the report of a customers and later stores it (report of a customer).

Module Specification

MODULE NAME: Generate report

PURPOSE: To generate the report of a customer

PSEUDOCODE:

FUNCTION generateReport

GET brief details of the customer

UPDATE details of the customer if required

Generate report of the customer

Store the generated report of a customer

END

INPUT PARAMETERS: Details of the customer

OUTPUT PARAMETERS: Store the report of a customer

GLOBAL PARAMETERS: Not assigned

LOCAL PARAMETERS: Not assigned

CALLS: Details of customers and staff

CALLED BY: Null

5. Conclusion

The coursework which had been assigned to us of Software Engineering consist of total of 20% of an overall module weight. In this coursework, we group members of this work (Ashmina Rai, Ashesh Rai, Sachin Dahal, Bibek Pradhan and Nishan Kadel) create a required analysis and a system for Sound Strong Music institute which is currently facing difficulties in registering forms and to record the number of customers by the staff. For this, we the members of our group created a system following different ERD, data flow diagram (Level 1 and Level 2) along with Structure chart.

Overall, the coursework has been really helpful to enhance the knowledge of the dataflow, diagram, structure chart and ERD. Similarly, the effort and dedication that every student has put toward the coursework is inevitable and would not have been possible to complete it in the times without the dedication and effort. Thus, we would like to thank every group members and teachers for guiding us through our coursework even in this situation.

6. References

Bell, D., 2005. Desgn. In: Fourth, ed. *Software Engineering for Students.* Harlow: Pearson Education Limited, p. 424.

Coursehero, 2020. *Dfd fragments.* [Online]
Available at: https://www.coursehero.com/file/p64aofv/DFD-Fragments-A-DFD-fragment-is-created-for-each-use-case-triggered-by-an-event/#:~:text=requirements%20information.-,DFD%20Fragments%20A%20DFD%20fragment%20is%20created%20for%20each%20use,each%20part%20of%20the%20s

GreeksforGeeks, 2020. *Levels in Data Flow Diagrams (DFD).* [Online]
Available at: https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/
[Accessed 18 10 2020].

Lucidchart, 2020. *What is an Entity Relationship Diagram (ERD)?.* [Online]
Available at: https://www.lucidchart.com/pages/er-diagrams

Meador, D., 2018. *What is Data Dictionary.* [Online]
Available at: http://www.tutorialspoint.com/What-is-Data-Dictionary
[Accessed 25 Dec 2020].

Nishadha, 2020. *Ultimate Entity Relationship Diagram Tutorial (ER Diagrams).* [Online]
Available at: https://creately.com/blog/diagrams/er-diagrams-tutorial/#:~:text=Entity%20relationship%20diagrams%20are%20used,their%20relationships%20with%20each%20other.&text=In%20the%20diagram%2C%20the%20information,attributes%20of%20a%20particular%20entity.

techopediA, 2020. *Process Specification.* [Online]
Available at: https://www.techopedia.com/definition/28636/process-specification#:~:text=Definition%20-%20What%20does%20Process%20Specification,regulatory%2Fengineering%20requirements%20and%20procedures

Valcheva, S., 2019. *Context and Level 1 Data Flow Diagram Examples With Explanation and Tutorial.* [Online]
Available at: http://www.intellspot.com/data-flow-diagram-examples/
[Accessed 26 July 2019].

Wondershare EdrawMax, 2020. *What is a Context Diagram - Explain with Examples.* [Online]
Available at: https://www.edrawmax.com/context-diagram/

Thank You