Table of Contents

1. Introduction 1

2. Group Tasks 2

2.1. Environmental model specification 2

2.1.1. Context diagram 2

2.1.2. Level 1 Data Flow Diagram 3

2.1.3. Level 2 Data Flow Diagram 4

2.2. Internal model specification for the system 5

2.2.1. Entity Relationship Diagram (ERD) 5

2.2.2. Entity Relationship Diagram (ERD) for whole system 6

2.3. Data Dictionary 7

2.3.1. Data dictionary for whole system 8

2.4. Process specifications (Pspecs) for elementary processes 10

2.4.1. Process A 10

2.4.2. Process B 10

2.4.3 Process C 11

2.4.4. Process D 11

2.5. Design specification 12

2.5.1. Structure Chart 12

2.6. Assignment diary 14

2.6.1. Assumptions 14

2.6.2. Group member responsibilities 16

2.6.3. Meeting Minute (Group meetings) 17

2.7. Data Flow Diagram 18

2.8. Module specification 18

3. Individual task 19

3.1. Register a customer 19
3.1.1. Environmental model specification 19
3.1.2. Internal model specification 19
3.1.3. Design specification 22
3.2. De-register a customer 24
3.2.1. Environmental model specification 24
3.2.2. Internal model specification 24
3.2.3. Design specification 27
3.3. Generate Report 29
3.3.1. Environmental model specification 29
3.3.2. Internal model specification 29
3.3.3. Design specification 30
3.4. To-Do List 32
3.4.1. Environmental model specification 32
3.4.2. Internal model specification 33
3.4.3. Design specification 35
3.5. Set of Task 37
3.5.1. Environmental model specification 37
3.5.2. Internal model specification 37
3.5.3. Design specification 39
3.6. Payment of a customer 41
3.6.1. Environmental model specification 41
3.6.2. Internal model specification 41
3.6.3. Design specification 42
4. Summary 44
5. References 45

Table of Figures

Figure 1 Context diagram for whole system 2
Figure 2 Level 1 DFD for whole system 3
Figure 3 Level 2 DFD for whole system 4
Figure 4 Entity relationship diagram for whole system 6
Figure 5 Structure chart hierarchy (tutorialspoint.com, 2019)..... 12
Figure 6 structure chart for whole system 13
Figure 7 Level 0/ Context level diagram to register a customer 19
Figure 8 Level1 DFD to register a customer..... 19
Figure 9 Level1 DFD to register a staff..... 20
Figure 10 Level2 DFD to register a customer..... 21
Figure 11 Structure chart for registration..... 22
Figure 12: Level 0 DFD or Context Level Diagram..... 24
Figure 13: Level 1 diagram for de-registration..... 24
Figure 14: Level 2 Diagram of Manage Customer Records. 25
Figure 15: Level 2 Diagram of Manage Staff Contract. 26
Figure 16: Structure chart for De-registration 27
Figure 17: Level 0 or Context Level Diagram 29
Figure 18: Level 1 DFD for generate report of a customer 29
Figure 19: Level 2 DFD for prepare report of a customer..... 30
Figure 20: Structure chart for generate report 30
Figure 21: Level 0/ Context level diagram for to-do list 32
Figure 22: Level-1 DFD for manage todo list..... 33
Figure 23: Level 2 DFD for manage to-do list..... 34
Figure 24: Structure Diagram for to-do-list 35
Figure 25: Level 0 or Context Diagram..... 37
Figure 26: Level 1 DFD for manage task set..... 37
Figure 27: Level 2 DFD for manage task set..... 38
Figure 28: Structure Diagram for Set of Task..... 39
Figure 29 Level 0/ Context level diagram for payment of a customer 41
Figure 30 Level1 DFD to generate list of customers whose fee is pending..... 41
Figure 31 Level1 DFD to prepare fee change notice..... 42
Figure 32 Structure chart for payment of a customer 42

1. Introduction

This is the first group coursework of the module ‘Software Engineering’ which has been worked on by our group of 5 members. For this coursework, we are required to analyze requirements as a manager for a Fitness GYM who are facing lots of problems in maintaining their records of customer and payment details. In order to complete this group coursework successfully and efficiently, we had to hold a few meetings to gather the ideas and thoughts of all our group members and implement them to form a well-managed system.

In order to reduce and solve the problem of the Fitness GYM a system is developed called **Fitness Application**. This system is used to add and update details of a customer, deregister a customer, check payment details of a customer, define set of tasks to be carried out, generate report of certain customers, facility to create and manage to-do lists, add staff details, de-register staff and much more. This system makes it convenient for the customers of the gym and also the staff to manage the customer and payment records. In order to analyze requirements for the system we have used Data Flow Diagrams, Structure chart diagram, Entity Relationship Diagram, Data Dictionary, Process specs and Module specs. To develop these utilities, draw.io tool has been used.

The main objective of this coursework is to demonstrate the knowledge of ‘Structured Software Engineering’. During this coursework, we learned to work in a small group and complete our tasks in a given time scale. Doing this coursework, we learned to share our ideas, divide our tasks, work as a team, increase the efficiency of our tasks and also learned to be accountable for the mistakes. This coursework has given us the knowledge on Structured software engineering and how we can efficiently analyze requirements and solve the problems in a well-structured manner. We, as a group gave our best in making this coursework meaningful, efficient and successful.

2. Group Tasks

2.1. Environmental model specification

2.1.1. Context diagram

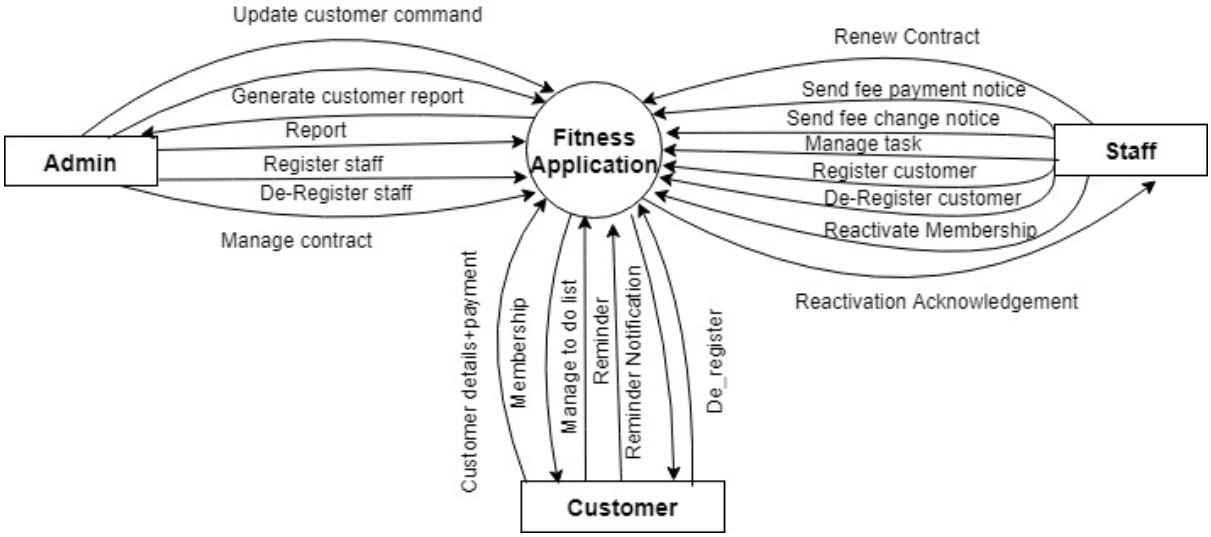

Figure 1 Context diagram for whole system

2.1.2. Level 1 Data Flow Diagram

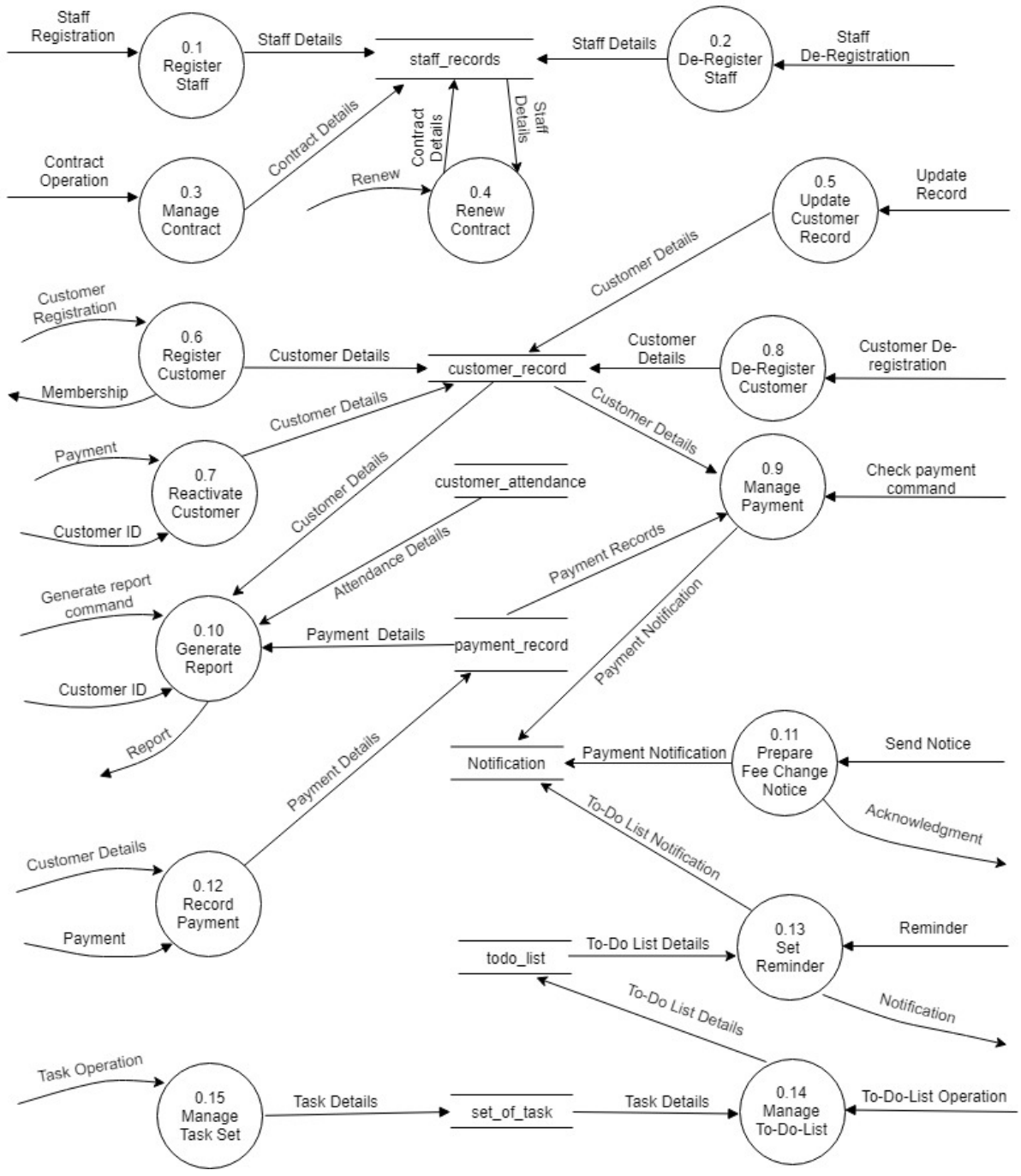

Figure 2 Level 1 DFD for whole system

2.1.3. Level 2 Data Flow Diagram

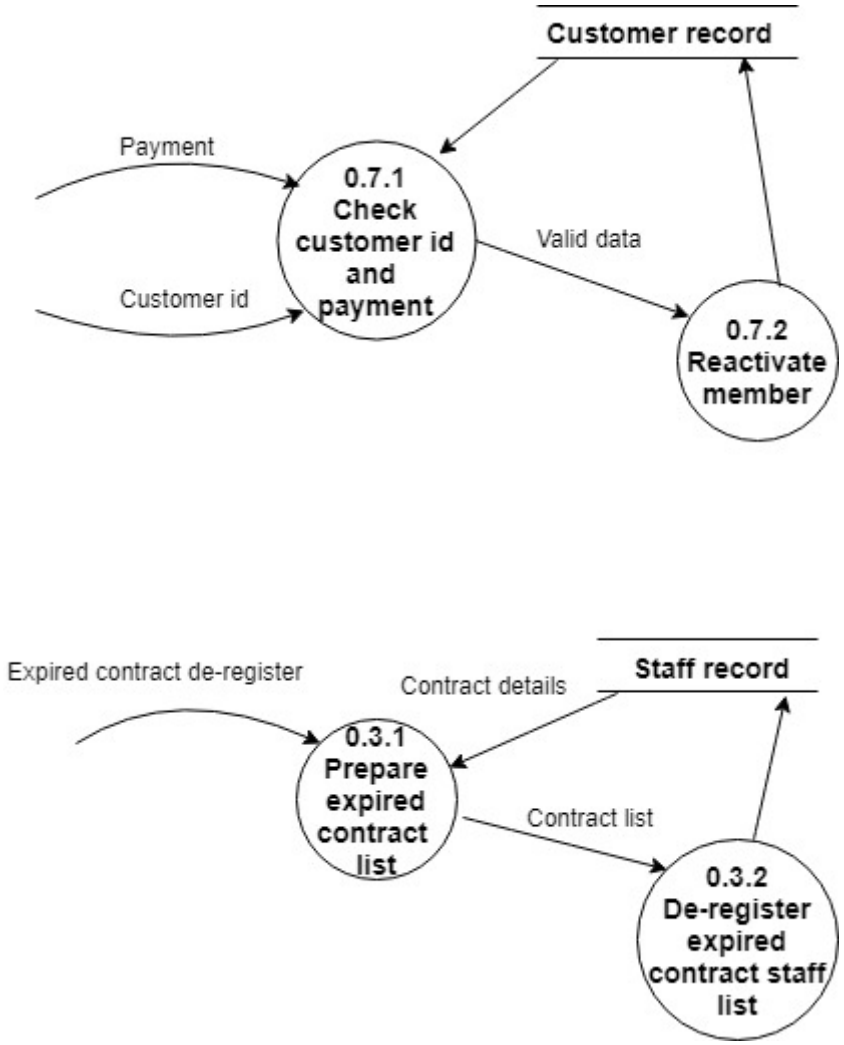

Figure 3 Level 2 DFD for whole system

2.2. Internal model specification for the system

2.2.1. Entity Relationship Diagram (ERD)

The graphical representation of an information system that shows the relationships among people, objects, places, concepts or events with in the system is an entity relationship diagram. Entity relationship diagrams provides a visual starting point for designing the database (searchdatamanagement.techtarget, 2019). Entity Relationship Diagram was proposed by Peter Chen in 1971 to create a uniform convention which can be used for relational database and network. He has aimed to use ER model as a conceptual modelling approach. (guru99, 2019) An ER diagram has three main components, they are:

a. **Entity**

   An entity is an object or component of data. It is represented as a rectangle in an ER diagram.

b. **Attribute**

   Attributes are the description property of an entity. It is represented as an oval in the ER diagram.

c. **Relationship:**

   Relationship shows the relation among the entities. A relationship is represented by diamond shape in ER diagram (beginnersbook, 2019).

2.2.2. Entity Relationship Diagram (ERD) for whole system

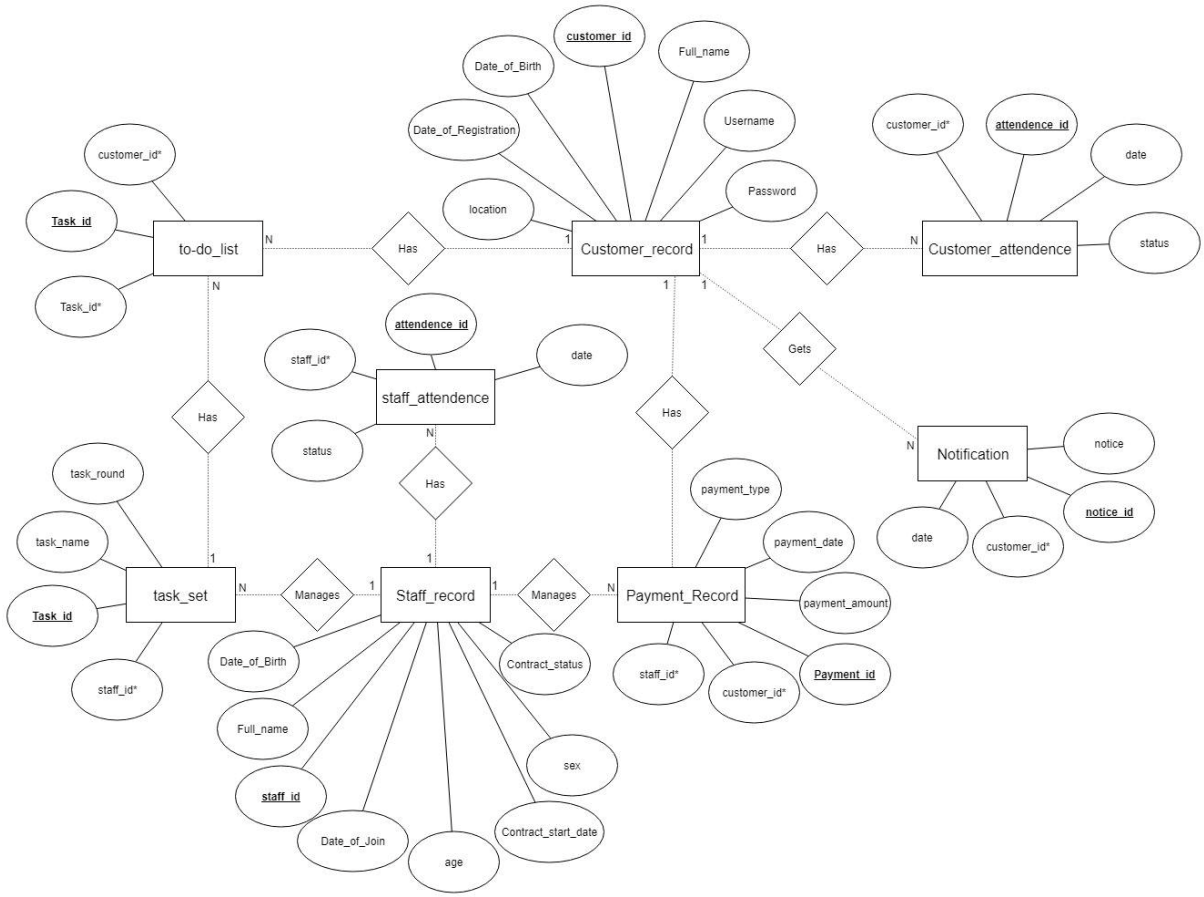

Figure 4 Entity relationship diagram for whole system

2.3. Data Dictionary

Data dictionary is the centralized collection of information about data which stores meaning and origin of the data, its relation with other data, data format for usage and many more. Data dictionary is also referenced as meta-data. (tutorialspoint, 2018) Data dictionary is also called Data Definition Matrix as it provides detailed information about the business data such as standard definitions of data elements, their meanings and allowable values. (bridging-the-gap, 2018) Data dictionary removes any chances of ambiguity and keeps the work of programmers and designers synchronized while using same object reference everywhere in the program. (tutorialspoint, 2018). When developing programs that use data model, a data dictionary can be consulted to understand where the data item fits in the structure, values that it contains and what the data item means in real world terms. (searchmicroservices, 2005)

2.3.1. Data dictionary for whole system

Update customer command = Command

Generate customer report = Command

Report = customer records + payment records + attendance

Staff registration = Command

Staff deregistration = Command

Manage contract = Command

Renew contract = Command

Send fee payment notice = Command

Send fee change notice = Command

Manage task = Command

Customer registration = Command

Customer deregistration = Command

Re-activate membership = Command

Reactivation acknowledgement = String

Customer details= Full Name + Username + Password + Date of Birth + Date of Registration + Location

Staff details= Full Name + Date of Birth+ Age+ Sex + Contract_start date + date join + contract status

Payment = Payment amount + Payment Date + Customer details+ [mobile_banking, credit_card]

Membership = Customer details + Payment

Manage to-do list = Command

Reminder = Command

Reminder notification = String

Contract operation= Command

Customer ID = Integer

Update records = Command

Check Payment command = Command

Task operation = Command

To-do operation = Command

Task details = Task ID+ Task Name+ Task rounds+ Staff ID

Task set = {Task details}*

To-do list = {Task ID+ Customer ID}*

Notification = {Notice ID+ Notice+ Customer ID+ Date}*

Staff record = {Staff ID+ Staff details}*

Customer record = {Customer ID+ Customer details}*

Payment record = {Payment ID+ Payment+ Staff ID}*

Customer attendance = {Customer ID+ status+ Date}*

2.4. Process specifications (Pspecs) for elementary processes

2.4.1. Process A

Number: 0.3.1

Name: Prepare expired contract list

Description: This process prepares a list of staffs whose contract has been expired.

Input Data Flow: expired contract list deregistration command+ contract details

Output Data Flow: expired contract list

Process Logic: The logic for this process is as follows-

i. Get contract details of all staffs from Staff records Database.
ii. Filter expired contracts and create a list.
iii. Expired contract list is passed to next sub process (0.3.2).

2.4.2. Process B

Number: 0.3.2

Name: De-register expired contract staff list.

Description: This process deregisters the staff whose contract has been expired.

Input Data Flow: expired contract list.

Output Data Flow: updated contract details

Process Logic: The logic for this process is as follows-

i. Expired contract list of staff is obtained from parent process.
ii. De-register all the staff in the list and update staff records.

2.4.3 Process C

Number: 0.7.1

Name: Check Customer ID and Payment

Description: This process checks the Customer ID and payment details.

Input Data Flow: Payment+ Customer ID+ Customer details

Output Data Flow: valid data

Process Logic: The logic for this process is as follows-

i. Staff supplies Customer ID.
ii. Customer ID is validated by checking it on Customer record database.
iii. If Customer ID is valid, then payment records associated with that Customer ID is queried.
iv. If valid payment record is formed, then valid data is passed to next sub process (0.3.2).

2.4.4. Process D

Number: 0.7.2

Name: Reactivate member

Description: This process reactivates membership of a customer.

Input Data Flow: valid data

Output Data Flow: updated customer details

Process Logic: The logic for this process is as follows-

i. Valid data is obtained from the parent process.
ii. Customer records database is updated with the received details and the gym membership of the customer is renewed.

2.5. Design specification

2.5.1. Structure Chart

Structure charts also known as module chart or hierarchy chart represent the structure and information about a program in visual form. It is prepared in Structured Design after creating DFD in Structured Analysis. The features of structure chart include the following:

- Separate program into modules
- Top-down hierarchy of modules
- Modules are linked with each other
- Depicts the flow of data, control and exceptions

(excelsoftware.com, 2019)

Structure chart aids in modular programming based approach where the complete system is coded as small independent interacting modules. Each module is responsible for doing one specific task. It shows the relation and interaction between the modules in a software. Structure chart is designed before programming a software. They don't express the actual programmatic logic of the software. It assists a software analyst to develop a software which meets the required objectives. (freetutes.com, 2019)

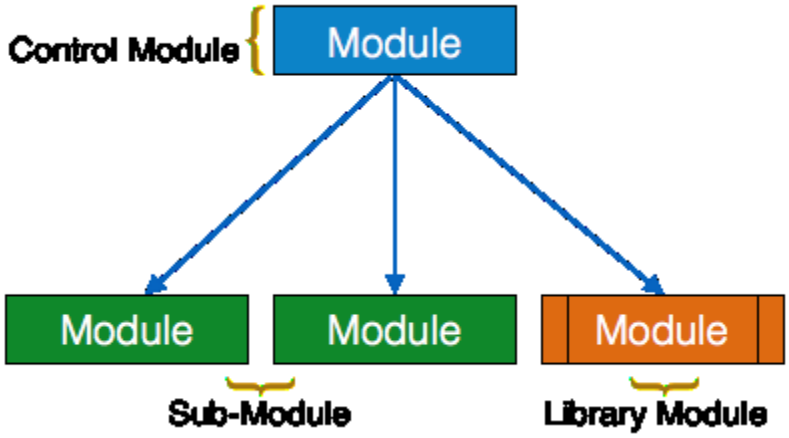

Figure 5 Structure chart hierarchy (tutorialspoint.com, 2019)

2.5.2. Structure chart for the whole system

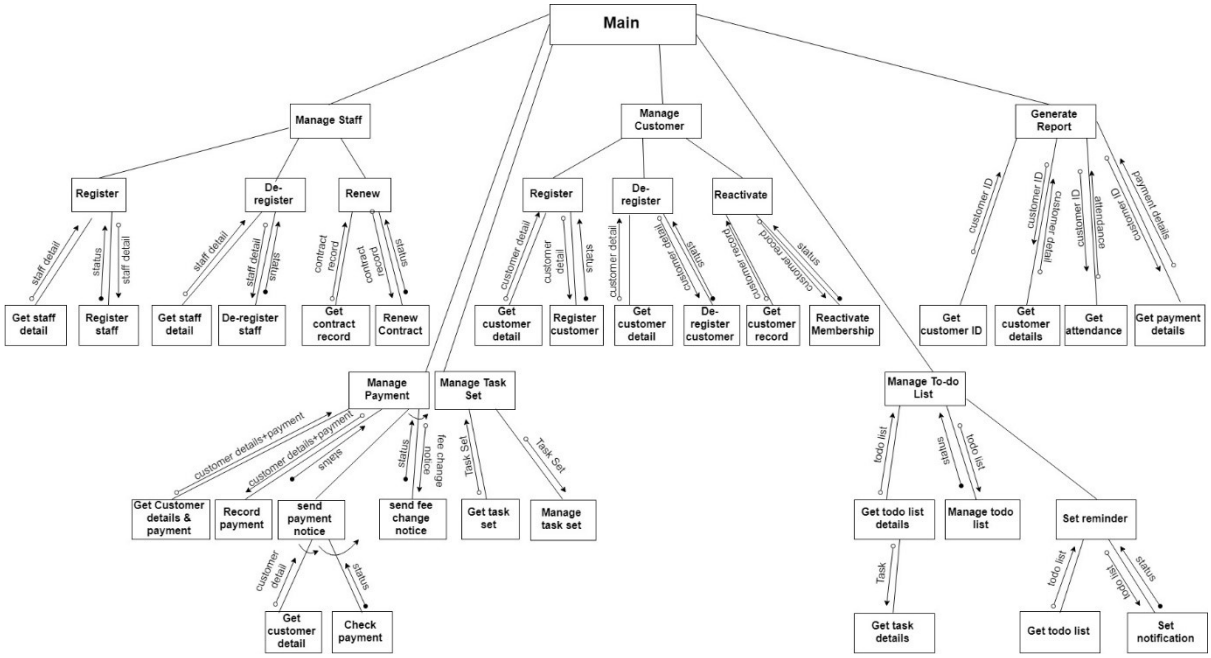

Figure 6 structure chart for whole system

2.6. Assignment diary

2.6.1. Assumptions

The provided specifications for the Fitness Application of the GYM company are analysed and worked upon to design the system meeting the required objectives. Few assumptions are made during the design process to enhance the logic and design of the system. Following are the assumptions made by the group members for the design of Fitness Application:

- Attendance system has been implemented for both employees and customers.
    - Attendance for customers keeps track of their active status in the gym and is used while generating a customer report.
    - Attendance of staff can be used to track the contract of a staff.
- The assumptions for database to store system data:
    - **customer_record :** This database stores all the personal details of the customers registered in the GYM.
    - **staff_record :** This database stores all the details of the staffs working in the GYM along with their contract details.
    - **customer_attendance :** This database holds daily attendance records of the customers.
    - **staff_attendance :** This database holds daily attendance records of the staffs working in the GYM.
    - **task_set :** This database stores all the tasks (daily tasks and tasks to be carried out at home) for the GYM customers. The tasks are defined by GYM staffs.
    - **todo_list :** This database stores all the todo lists created by the customers. The todo lists are created by referencing the task sets defined by GYM staffs on task_set database.
    - **payment_record:** This database stores all the monthly payments done by the customers for their GYM membership.
    - **notification :** This database stores all the notifications/alerts that are dispatched to staffs/customers in the Fitness Application for todo list reminder, fee change notice, fee payment notice etc.

Furthermore, following are the requirements collected by the team from the provided specification based on which the above assumptions are made:

- Maintain record of all the staff and customers.
- Customers must register to join the GYM.
- Payment if required before registration.
- Automatically de-register a customer after 30 days of inactivity.
- GYM staff can reactivate membership of an inactive customer.
- Customer Details
    - Full name of customer
    - Date of birth
    - Username
    - Password
    - Location
    - Date of registration
    - Payment amount
    - Payment type
- Staff maintains the payment details of the customer.
- Staff creates daily task to be carried out at the GYM and tasks to be carried out at home.
- Admin can generate report of customers.
- Customers can create to-do lists based on the tasks defined by GYM staffs.
- Customers can create to-do lists for both home tasks and daily tasks of the GYM.
- Customers can set reminder, update their to-do list, read the list and delete it.
- Staff can update records of a customer.
- Membership reactivation involves re-registration with payment.
- Staff can create pending fee customer list and inform them about the fee payment deadlines.
- Admin can register a staff.
- Admin can de-register a staff.
    - If staff leaves before termination of the contract then de-register the staff and restrict system usage.
    - After termination of contract, provide 10 days of buffer time for contract extension. Automatically de-register staff if buffer time exceeds.
- Customers can de-register themselves in case they are planning to leave the GYM.

2.6.2. Group member responsibilities

Each group member did an active collaboration while working on the design of the system. Group members helped each other to review their individual task and also discussions and interactions were involved about the whole system so that each and every group member understands the complete workflow of the system. Following are the responsibilities listed for individual group members:

Group Member	Responsibilities
	• Work on individual task for **De-registration of customer**. • Work on Entity Relationship diagram literature. • Design the Entity Relationship diagram. • Contributed to assumptions part of the report.
	• Work on individual task for **Generate report of a customer**. • Work on Module Specification literature. • Mock the designs of DFD level 0 and level 2 of the whole system. • Contributed to summary portion of the report documentation.
	• Work on individual task for **Create task set and to-do list**. • Work on Data Flow Diagram literature. • Mock the design of DFD level 1 of the whole system. • Contributed to summary portion of the report documentation.
	• Work on individual task for **Check payment of a customer**. • Work on Structure Chart literature. • Mock the design of Structure Chart of the whole system. • Contributed to assignment diary portion of the report. • Provide direction, instructions and guidance to team members. • Monitor the work progress and re-align the project on track.
	• Work on individual task for **Registration of Customers and Staffs**. • Work on Data Dictionary literature. • Worked on Data Dictionary of the whole system. • Worked on Process Specifications of the whole system. • Worked on Introduction portion of the report documentation.

2.6.3. Meeting Minute (Group meetings)

Date	Time	Location	Discussions
12/12/2018	8:30 AM to 10: 00 AM	London Block	We analysed the problem of the course work on the first day and divided the work among the group members.
13/12/2018	10: 00 AM to 11: 30 AM	UK Block	On the second day work plan was devised and requirements analysis was done. We discussed the Fitness Application requirements in detail.
17/12/2018	10: 00 AM to 11: 00 AM	Nepal Block	All of the group members worked on the group task to design the DFD and Data Dictionary for the entire system. We met our module lecturer and cleared our doubts about the specifications.
18/12/2018	2: 00 PM to 3: 30 PM	London Block	We discussed and corrected errors in the design. Also, we started to work on the Structure Chart for overall system.
19/12/2018	8: 30 AM to 10: 00 PM	UK Block	We met the module lecturer and discussed the problems we faced on the concepts of DFD and Structure Chart. We also discussed about the structure and formatting of the report.
20/12/2018	7: 00 AM to 8: 30 PM	Nepal Block	Each group member worked on their individual task. We reviewed each other's design implementations, discussed and improvised it.
21/12/2018	12:00 PM to 1:30 PM	Nepal Block	We designed the diagrams and charts.
26/12/2018	8:30 PM to 10:00 PM	UK Block	We assembled the whole report by combining individual tasks.
06/01/2019	8:00 PM to 9:00 PM	UK Block	We met the module lecturer and discussed about the problems we faced.
08/01/2019	2:00 PM to 5:00 PM	UK Block	We did proof reading, detailed review and report formatting before handing in the work.

2.7. Data Flow Diagram

Data Flow Diagram is a major design which graphically represents the visual information flow of data in an information system. Graphical representation makes a good communication between user and system designer (Bruza & Weide, 1989). There are three levels of data flow diagram. Some of them are Level 0 or context level diagram, Level 1 and Level 2. (Hathaway & Hathaway, 2016) Data flow diagram demonstrates how the information enters and leaves the system and the changes made by entering and leaving the system in the information and where the information is stored.

2.8. Module specification

“Module specification is a statement of effect that a software model is required to achieve”. (Encyclopedia, 2016) It can be used by implementer of the module and also by the user of the module. It is used by implementer of the module because it provides exact statement required for the module. It is also used by the user of the model because it gives the statement about what module provides. There are different techniques developed for module specification like functional specification, algebraic specification and some other (Encyclopedia, 2016).

3. Individual task

3.1. Register a customer

3.1.1. Environmental model specification

3.1.1.a. Context level diagram

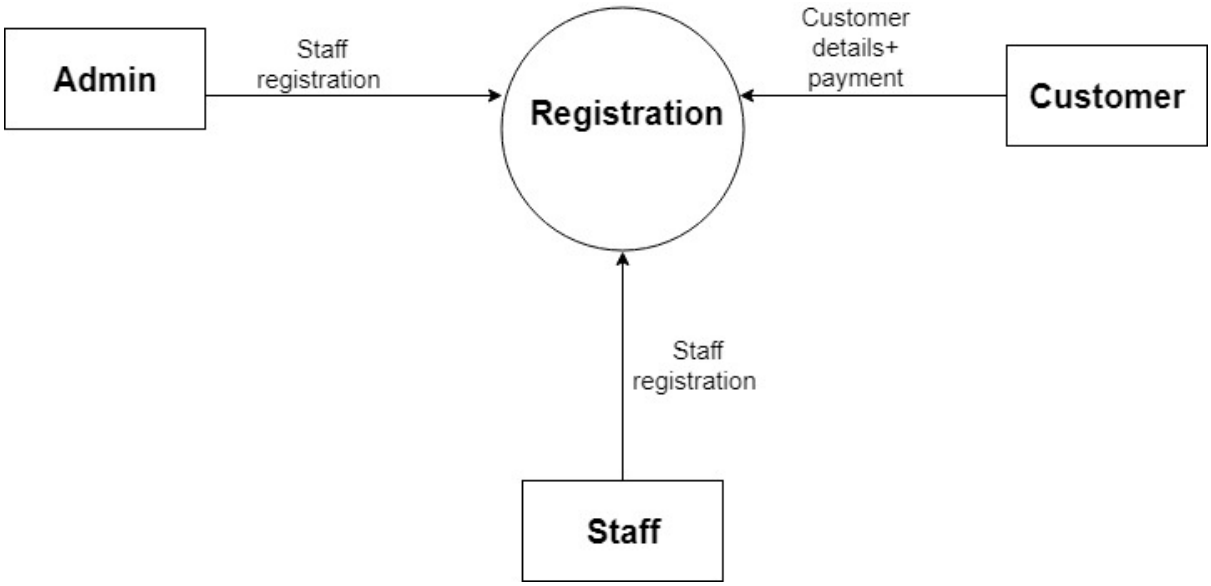

Figure 7 Level 0/ Context level diagram to register a customer

3.1.2. Internal model specification

3.1.2.a. Level 1 DFD

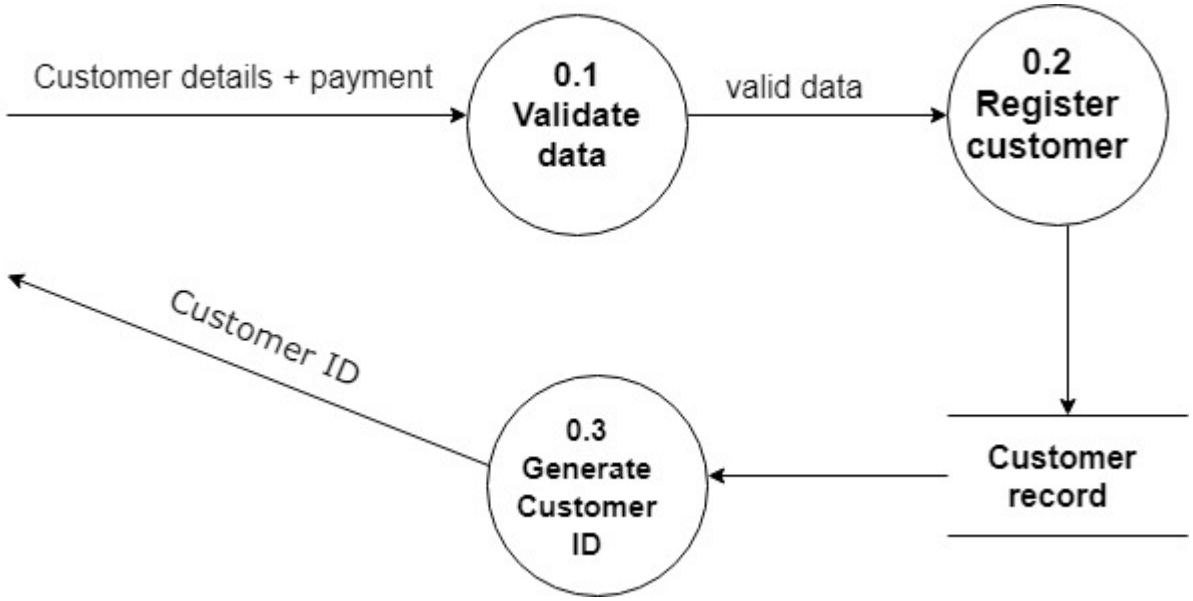

Figure 8 Level1 DFD to register a customer

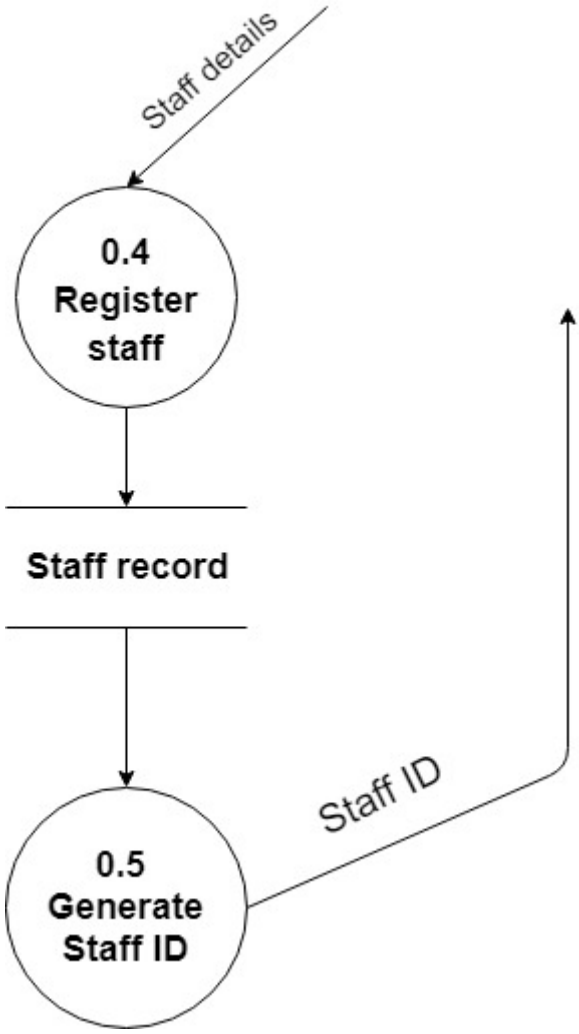

Figure 9 Level1 DFD to register a staff

3.1.2.b. Level 2 DFD

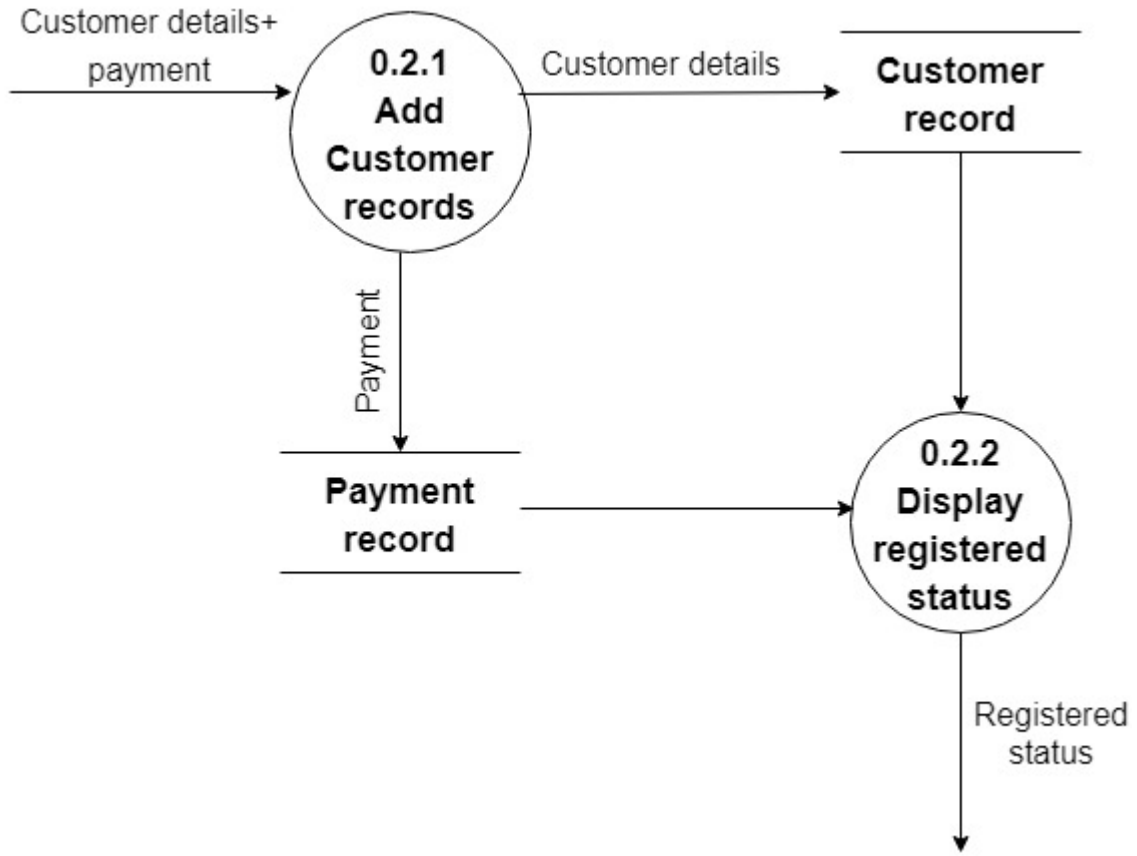

Figure 10 Level2 DFD to register a customer

3.1.3. Design specification

3.1.3.a. Structure Chart

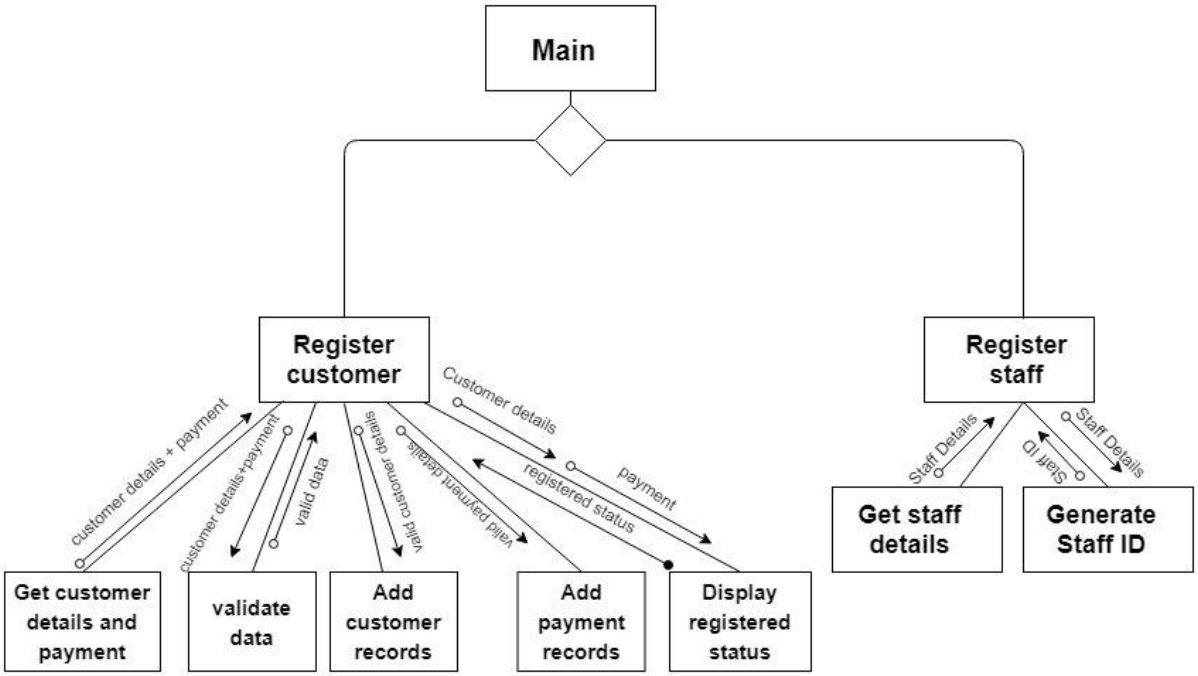

Figure 11 Structure chart for registration

3.1.3.b. Module specs

MODULE NAME: Register Customer

PURPOSE: This module obtains customer and payment details from the user and adds these details to the Customer record database and Payment record database respectively.

PSEUDOCODE:

```
var full_name = input("Enter Full Name")
var username=input("Enter username")
var password=input("Enter password")
var location=input("Enter location")
var DOB=input("Enter date of birth")
var DOR=input("Enter date of registration")
var payment_amount=input("Enter payment amount")
var payment_type=input("Enter payment type")
var payment_date= input("Enter payment date")
if(validate (full_name,username,password, location, DOB,
DOR,payment_amount, payment_type,payment_date)){
Var member_ID= DB.customer_record(full_name, username, password, DOB,
location, DOR) //insert customer record into db and return member id
DB.payment_record(member_ID, payment_amount, payment_type,
payment_date)
Display("registration successful")
Display("member ID:"+ member_ID)
}
```

INPUT PARAMETERS : customer_list

OUTPUT PARAMETERS : member_ID

GLOBAL VARIABLES : DB

LOCAL VARIABLES : notice, customer_id, date

CALLS: ***Get Customer details and payment, validate data, Add customer records, Add payment records, Display registered status.***

C*ALLED BY:** ***Main

3.2. De-register a customer

3.2.1. Environmental model specification

3.2.1.a. Context level diagram

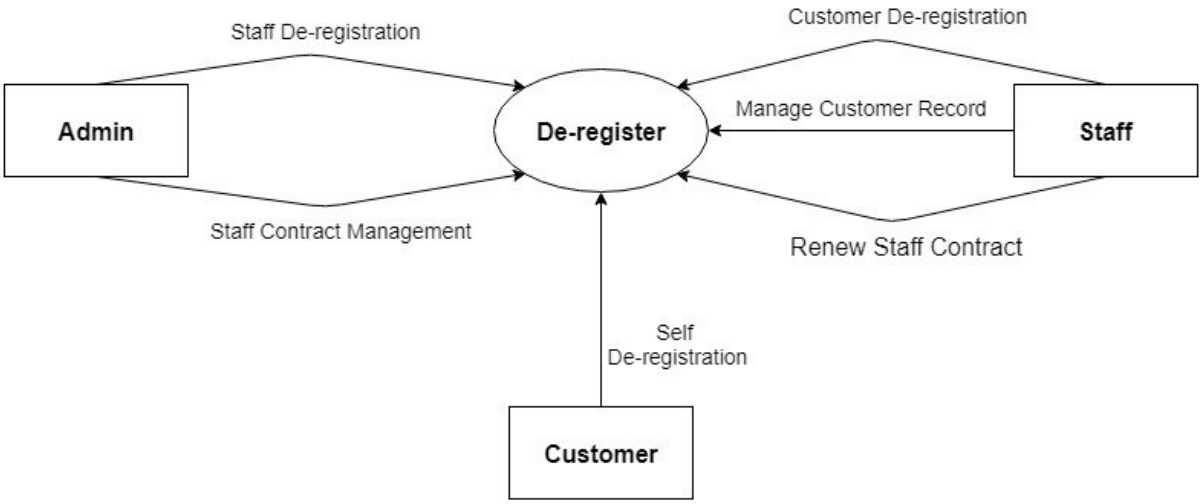

Figure 12: Level 0 DFD or Context Level Diagram

3.2.2. Internal model specification

3.2.2.a. Level 1 DFD

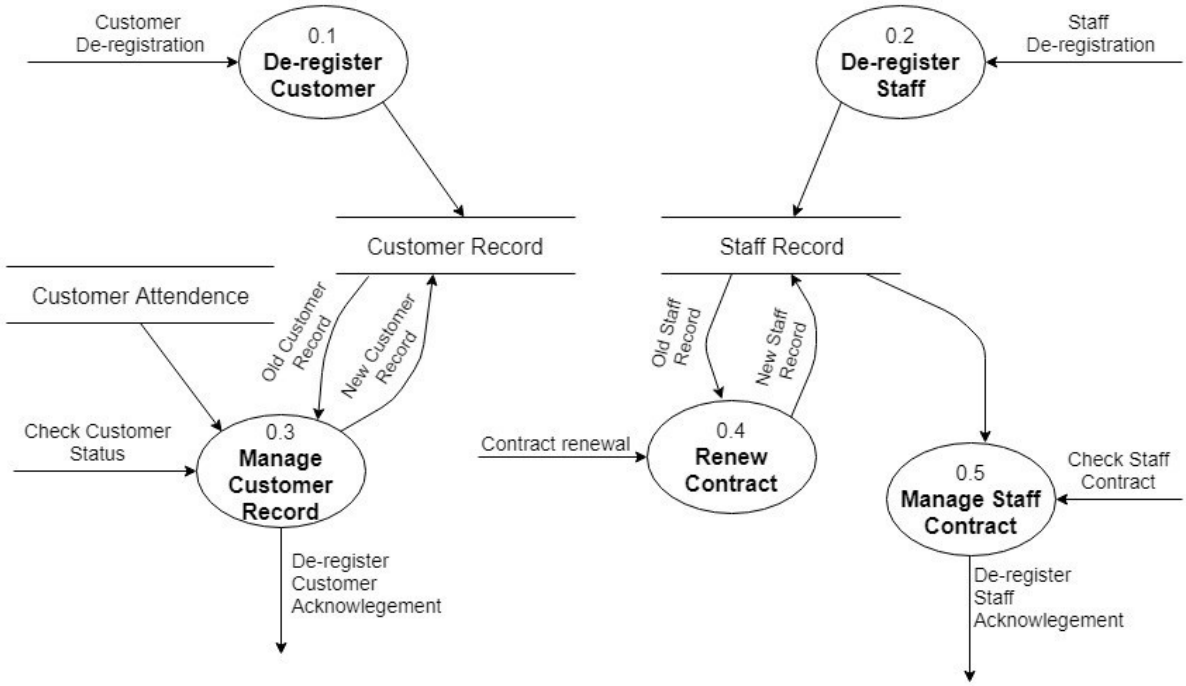

Figure 13: Level 1 diagram for de-registration

3.2.2.b. Level 2 DFD

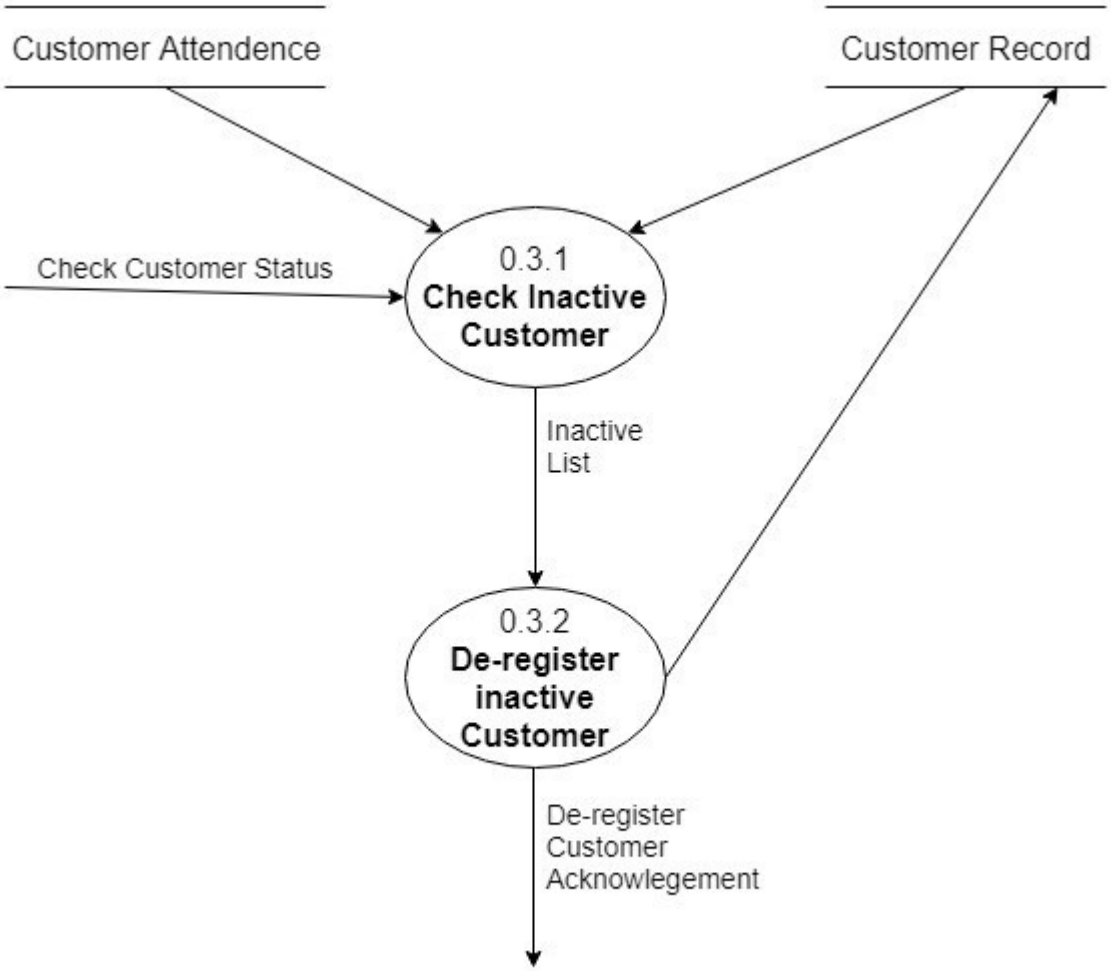

Figure 14: Level 2 Diagram of Manage Customer Records.

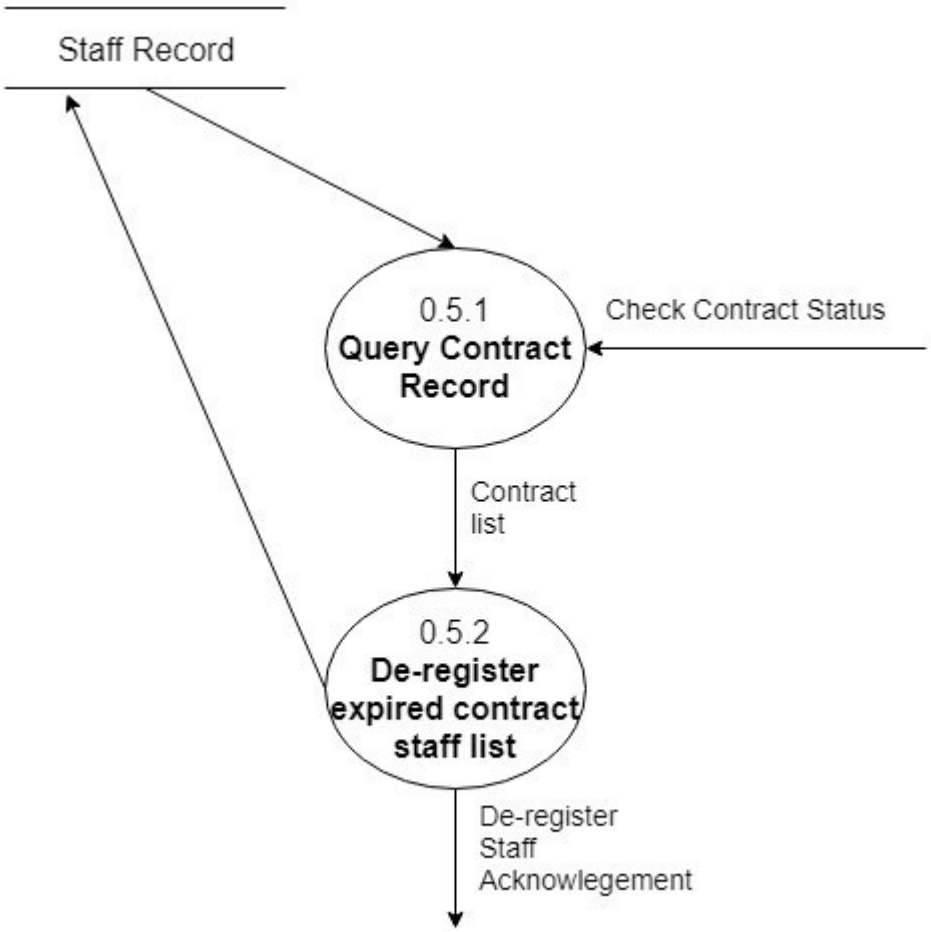

Figure 15: Level 2 Diagram of Manage Staff Contract.

3.2.3. Design specification

3.2.3.a. Structure Chart

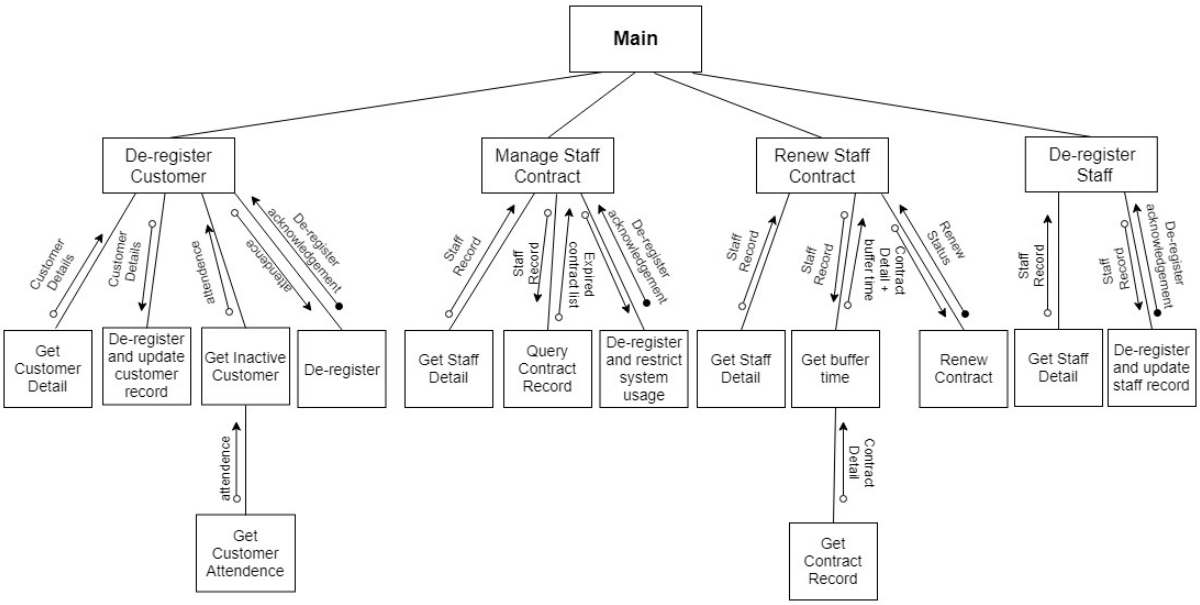

Figure 16: Structure chart for De-registration

3.2.3.b. Module Specs:

MODULE NAME: De-registration

PURPOSE: This module obtains staff details from the user and de-registers and updates the staff who are planning to leave the gym.

PSEUDOCODE:

```
DO
var staff_record = DB.getStaffRecord()
var staff_id = staff_record['id']
DB.de_register(staff_id)
DISPLAY('Deregistration Successful')
END DO
```

INPUT PARAMETERS : staff_record

OUTPUT PARAMETERS : staff_record

GLOBAL VARIABLES : DB

LOCAL VARIABLES : staff_record, staff_id

CALLS: ***Get Staff Detail, De-register and update staff record***

CALLED BY: ***Main***

3.3. Generate Report

3.3.1. Environmental model specification

3.3.1.a. Context level diagram

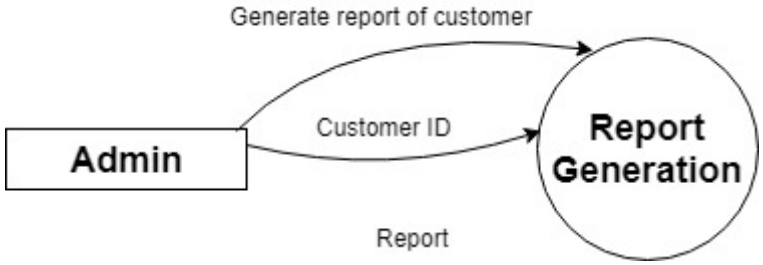

Figure 17: Level 0 or Context Level Diagram

3.3.2. Internal model specification

3.3.2.a. Level 1 DFD

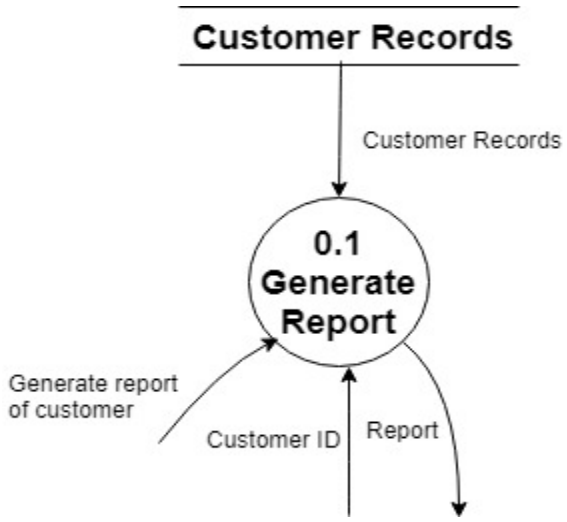

Figure 18: Level 1 DFD for generate report of a customer

3.3.2.b. Level 2 DFD

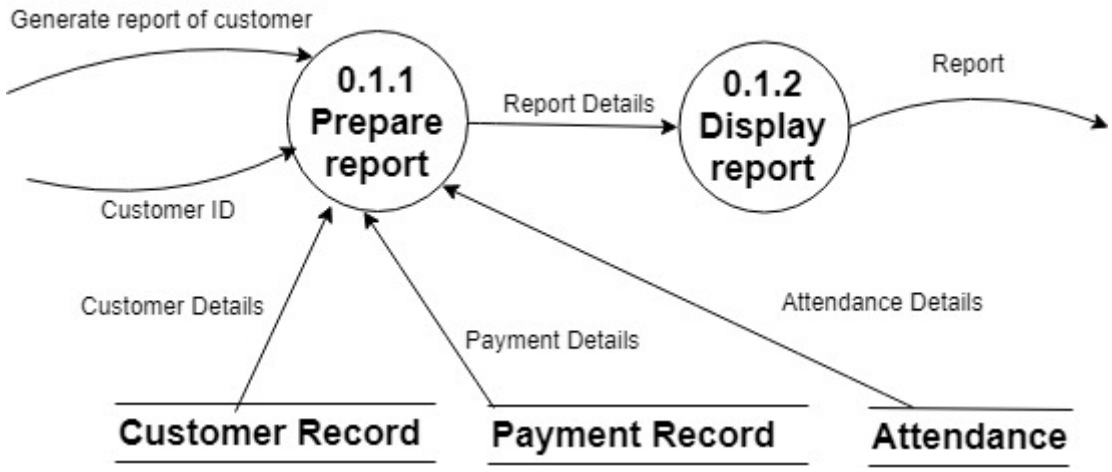

Figure 19: Level 2 DFD for prepare report of a customer

3.3.3. Design specification

3.3.3.a. Structure Diagram

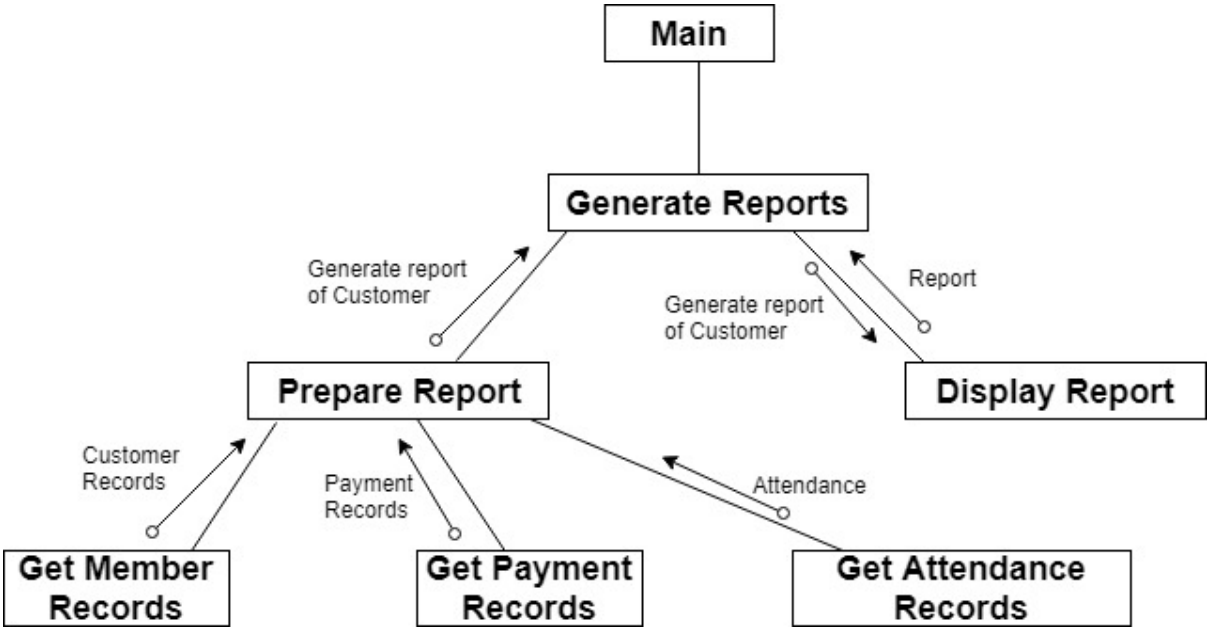

Figure 20: Structure chart for generate report

3.3.3.b Module Specs

Module Name: Generate Report

Purpose: The purpose of this module is to generate the report of the customer. To create the report, it collects the data from customer records, payment records and attendance database.

Pseudocode:

```
DO
 var customer_record=DB.get_customer_records()
var payment_record=DB.get_payment_records()
var attendance =DB.get_attendance()
var customer_id = customer_records['customer_id']
var report = generate_report(customer_record, payment_record, attendance)
Display(report)
END DO
```

Input Parameters: customer_records

Output Parameters: customer_records

Global Variable: DB

Local Variable: customer_record, payment_record, attendance, customer_id, report

Calls: ***Get customer records, payment record, attendance and generate report of the customer***

Called By: ***Main***

3.4. To-Do List

3.4.1. Environmental model specification

3.4.1.a. Context level diagram

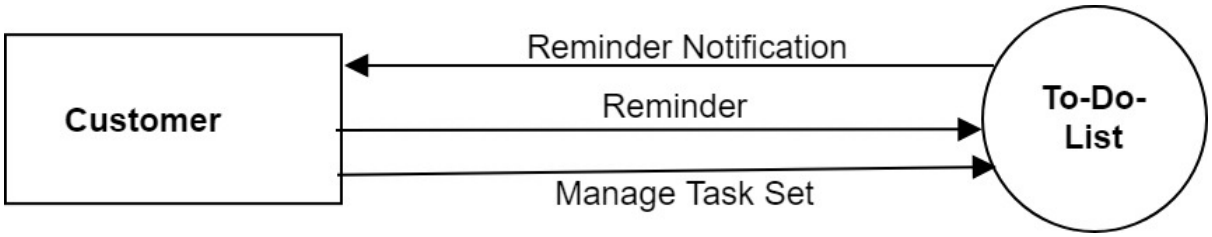

Figure 21: Level 0/ Context level diagram for to-do list

3.4.2. Internal model specification

3.4.2.a. Level 1 DFD

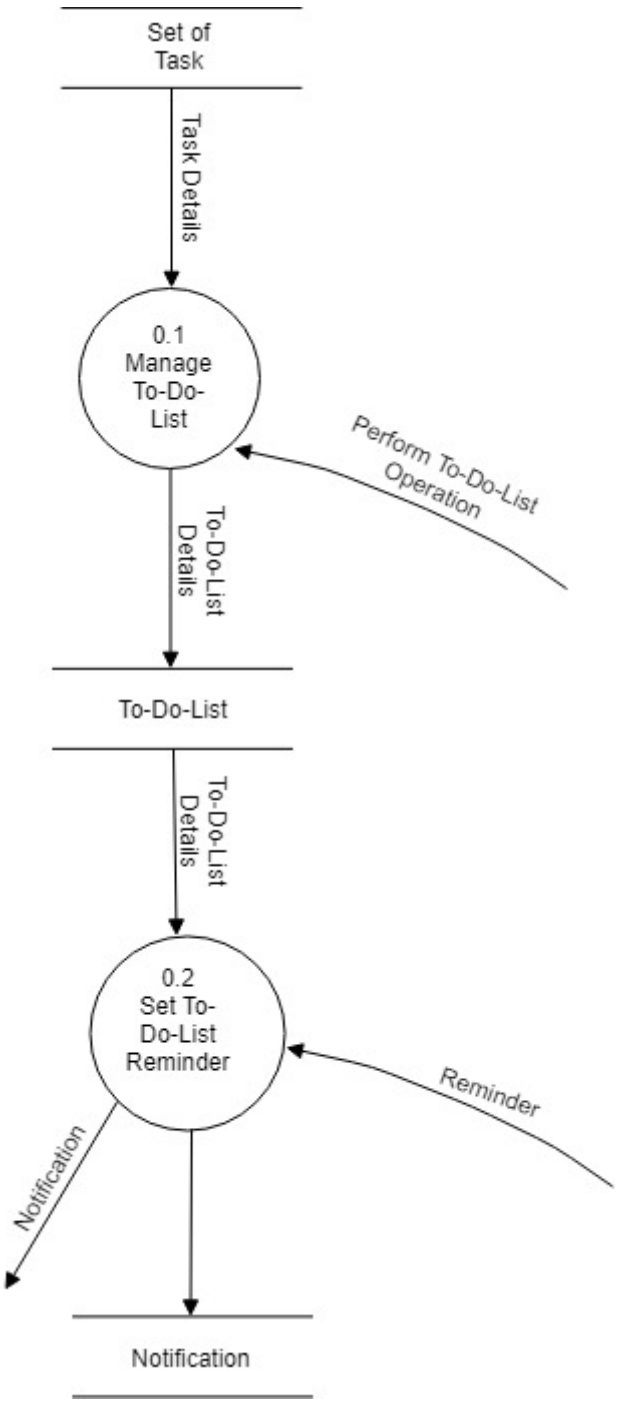

Figure 22: Level-1 DFD for manage todo list

3.4.2.b. Level 2 DFD

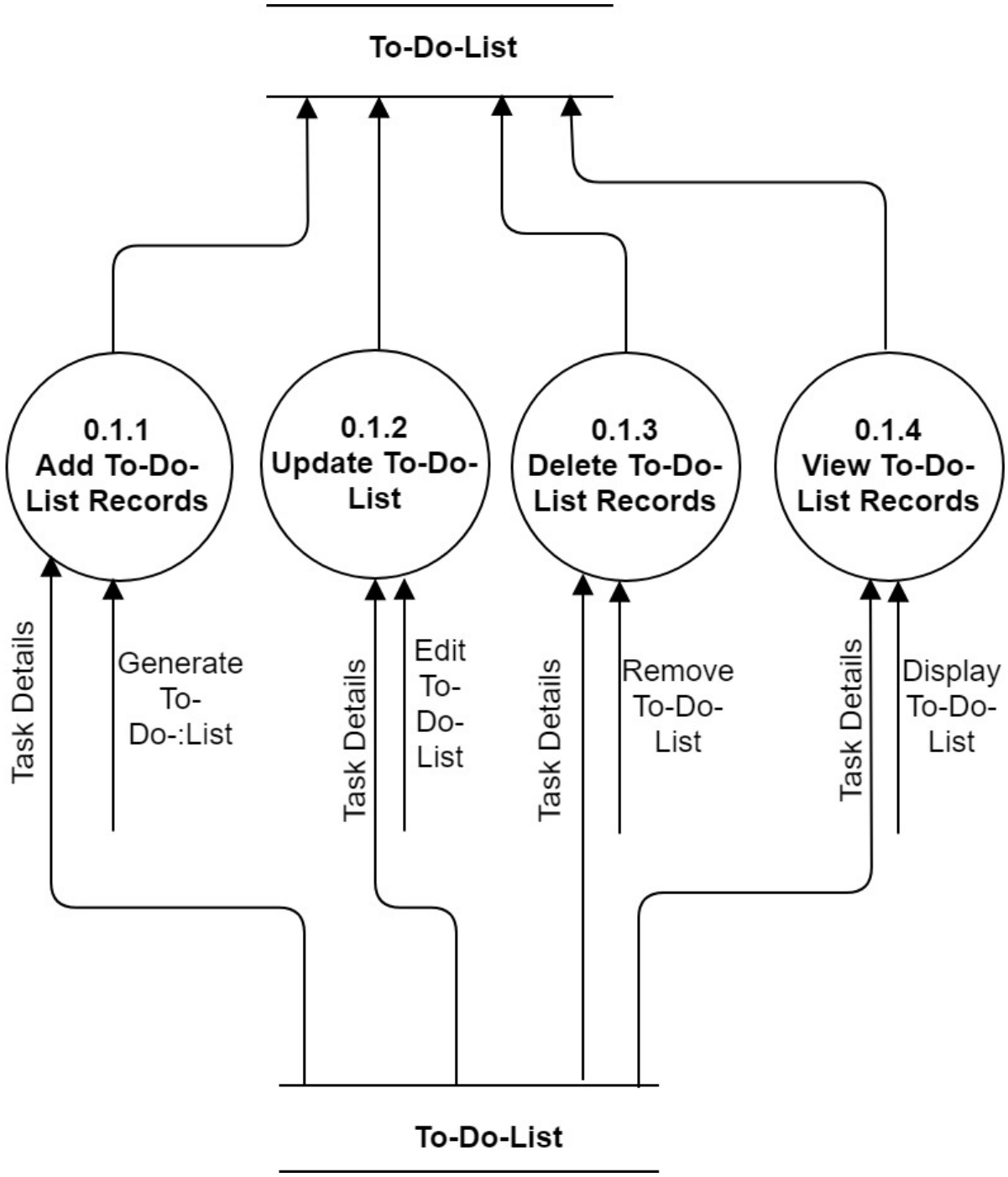

Figure 23: Level 2 DFD for manage to-do list

3.4.3. Design specification

3.4.3.a. Structure Chart

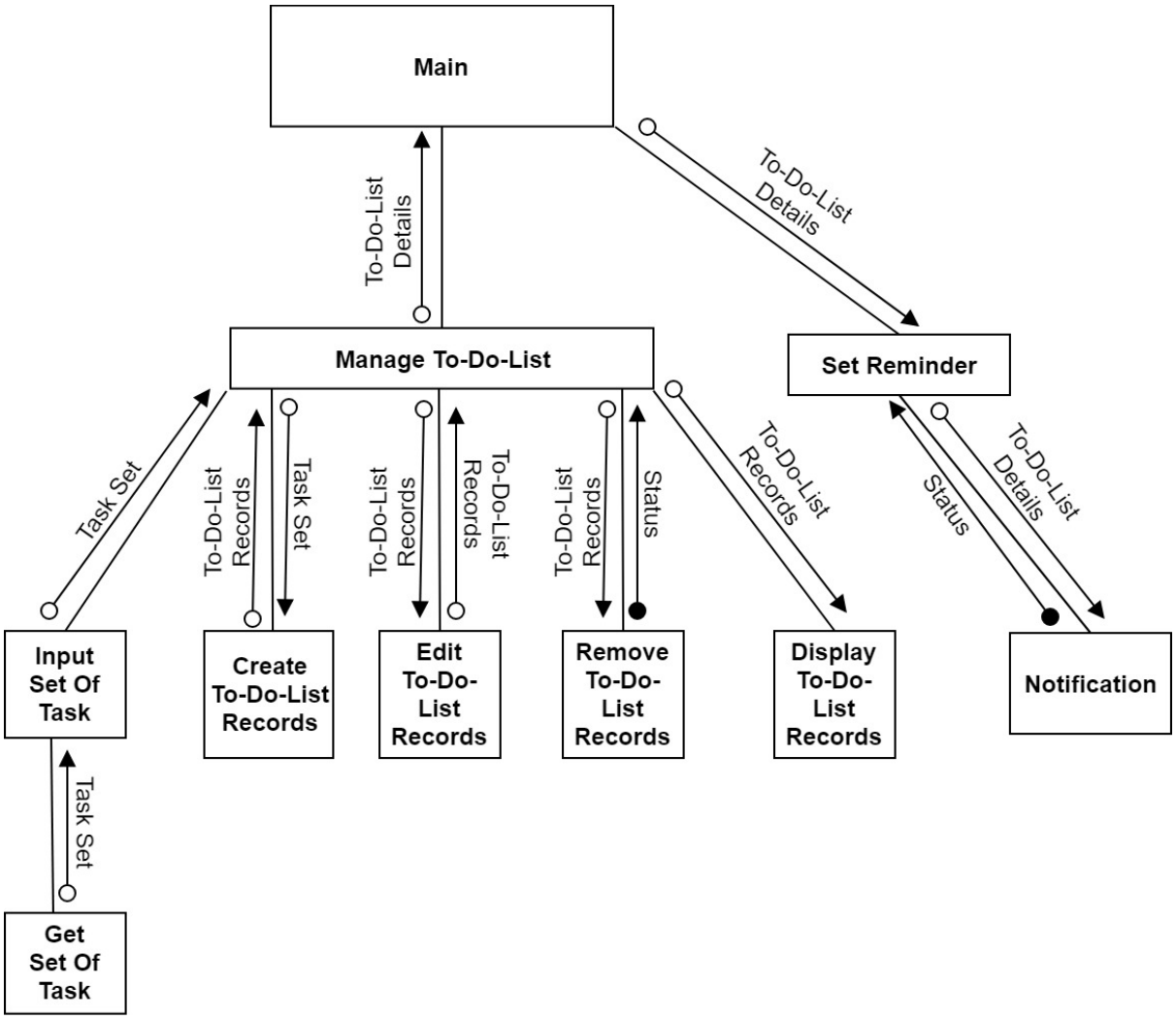

Figure 24: Structure Diagram for to-do-list

3.4.3.b. Module specs

MODULE NAME: Set Reminder

PURPOSE: This module sets the reminder for the customer to perform the set of tasks. It also provides the details of the to-do-list which needs to be achieved in a certain period of time.

PSEUDOCODE:

```
var to_do_list_details = DB.get_customer_to_do_list()
var notice = 'It's time to do 20 push ups.'
var customer_id = customer_to_do_list['customer_id']
var date = '2019/01/01'
NOTIFICATION.add_notification(notice, customer_id, date)
```

INPUT PARAMETERS :

OUTPUT PARAMETERS : to_do_list_details

GLOBAL VARIABLES : DB

LOCAL VARIABLES : notice, customer_id, date

CALLS: ***Notification***

CALLED BY: ***Main***

3.5. Set of Task

3.5.1. Environmental model specification

3.5.1.a. Context Level Diagram

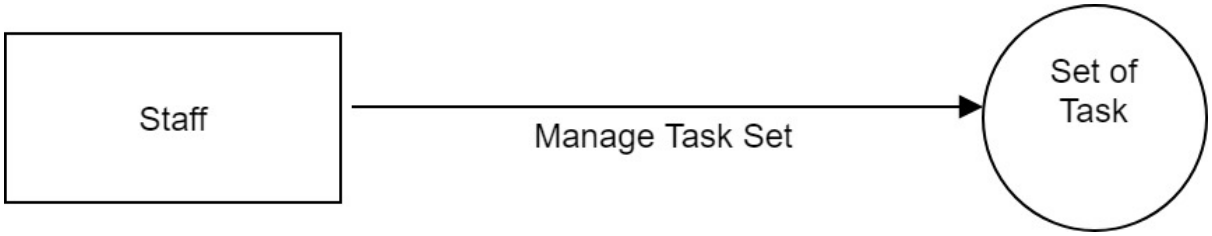

Figure 25: Level 0 or Context Diagram

3.5.2. Internal model specification

3.5.2.a. Level 1 DFD

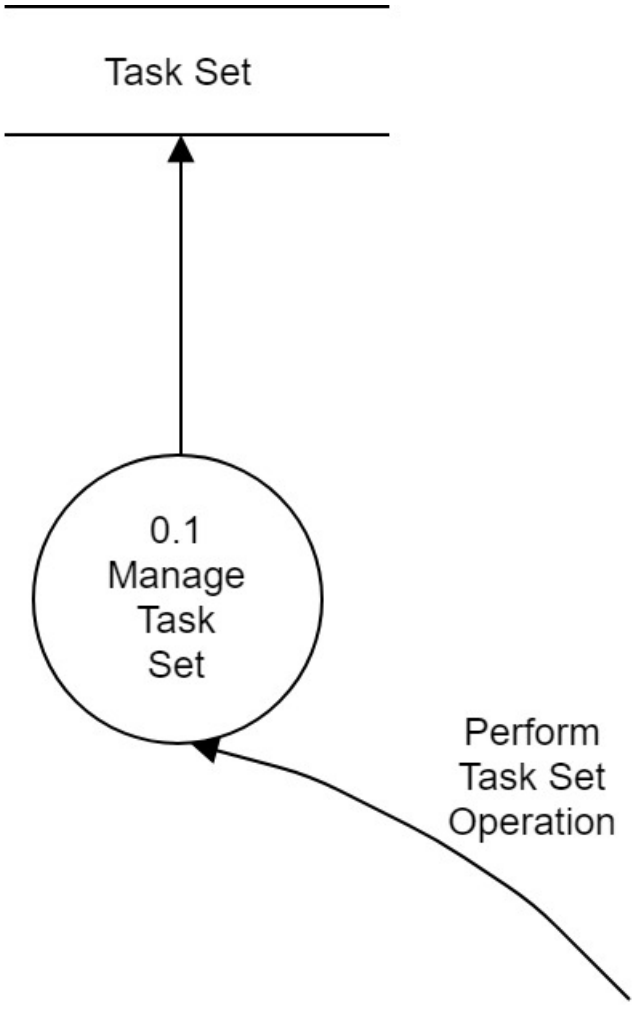

Figure 26: Level 1 DFD for manage task set

3.5.2.b. Level 2 DFD

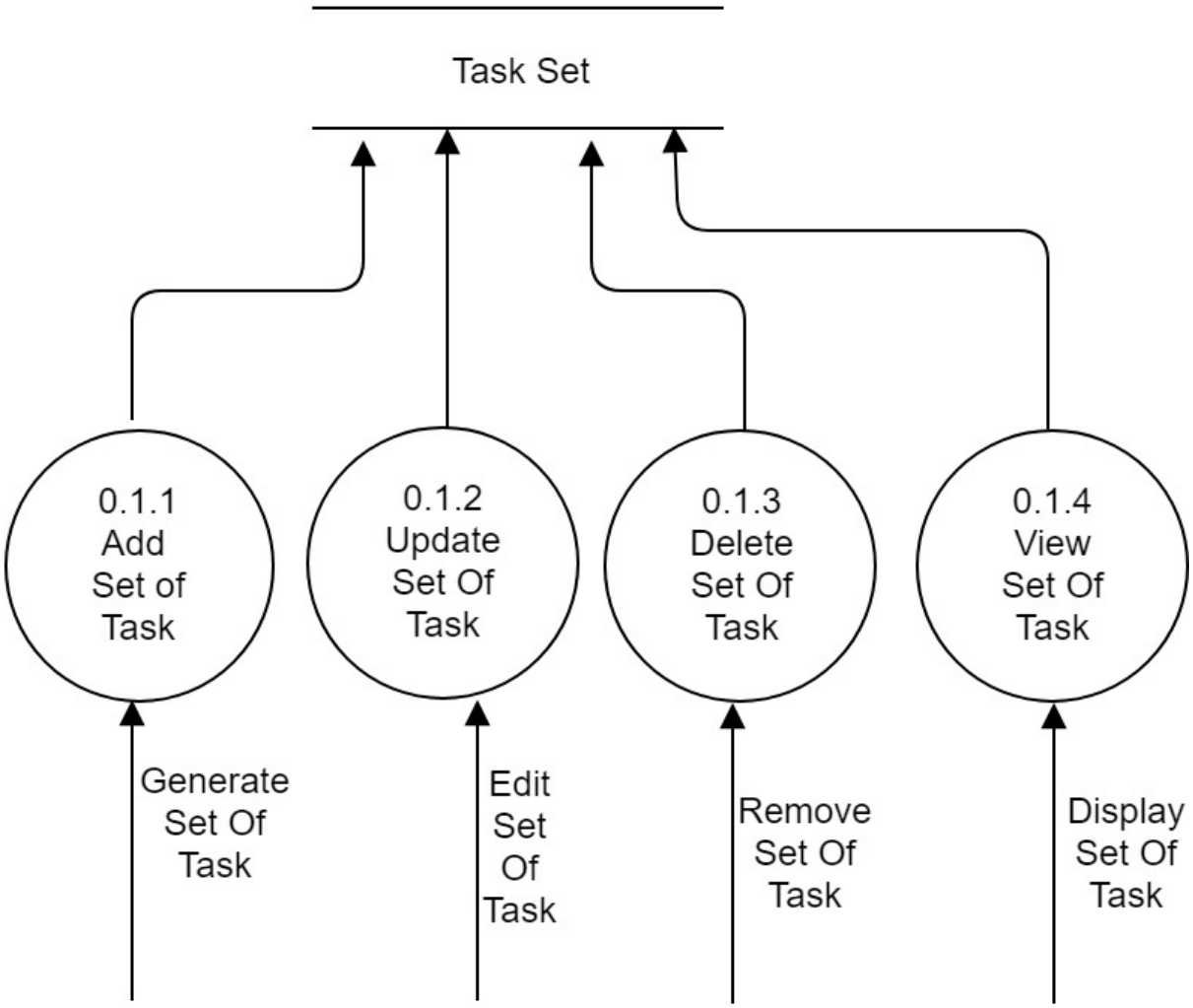

Figure 27: Level 2 DFD for manage task set

3.5.3. Design specification

3.5.3.a. Structure Diagram

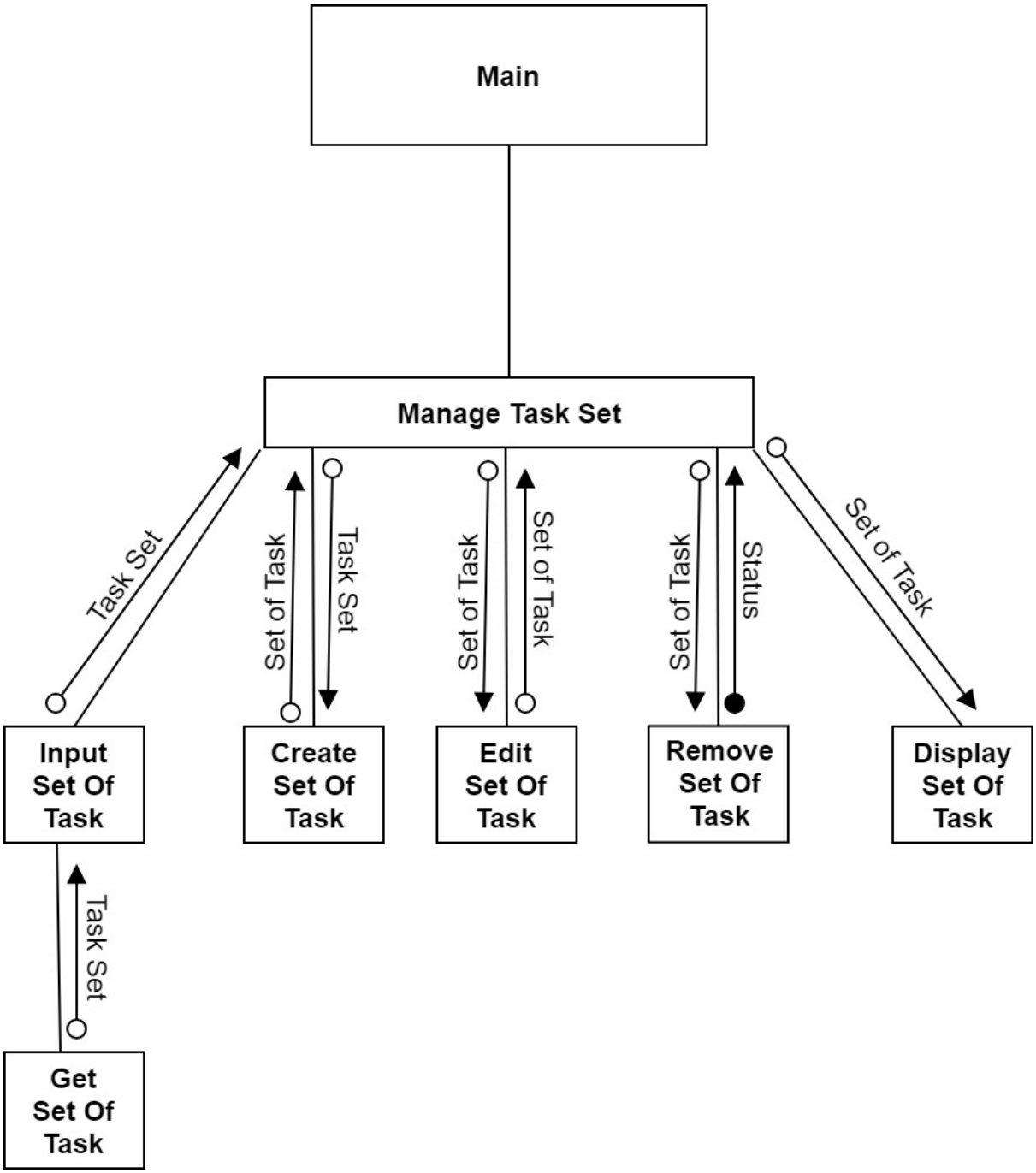

Figure 28: Structure Diagram for Set of Task

3.5.3.b. Module specs

MODULE NAME: Set of Task

PURPOSE: This module performs task set operation and then manage the task set and send them to the database named as Task Set.

PSEUDOCODE:

```
var task_set_details = DB.get_customer_task_set()
var customer_id = customer_task_set['customer_id']
var date = '2019/01/01'
Create.add_set_of-task(create, customer_id,set_of_task, date)
Edit.edit_set_of-task(edit, customer_id,set_of_task, date)
Display.print_set_of-task(Display, customer_id,set_of_task, date)
```

INPUT PARAMETERS : set_of_task
OUTPUT PARAMETERS : set_of_task
GLOBAL VARIABLES : DB
LOCAL VARIABLES : Display, customer_id,set_of_task, date
CALLS: Manage Task Set
CALLED BY: ***Main***

3.6. Payment of a customer

3.6.1. Environmental model specification

3.6.1.a. Context level diagram

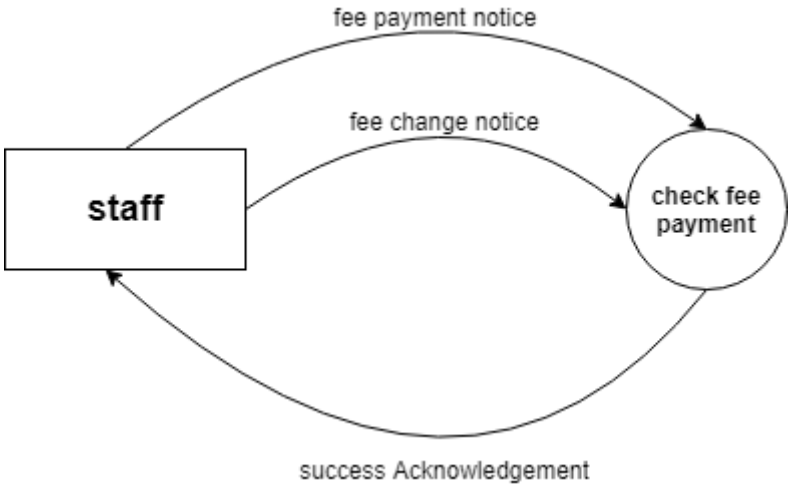

Figure 29 Level 0/ Context level diagram for payment of a customer

3.6.2. Internal model specification

3.6.2.a. Level 1 DFD

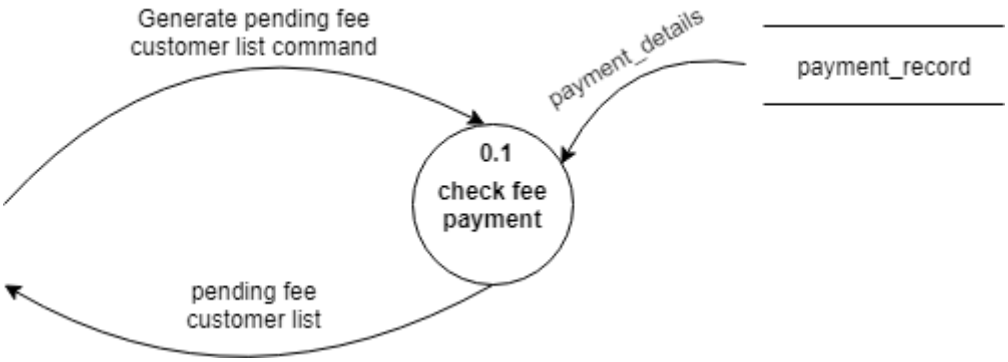

Figure 30 Level1 DFD to generate list of customers whose fee is pending

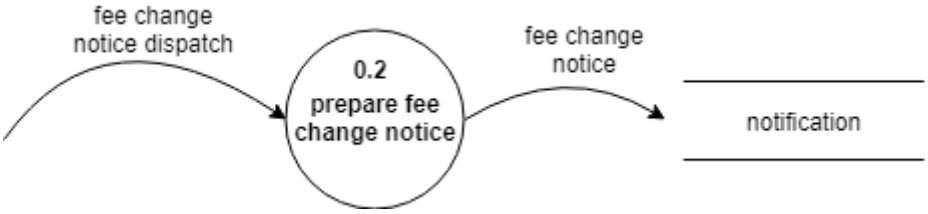

Figure 31 Level1 DFD to prepare fee change notice

3.6.3. Design specification

3.6.3.a. Structure Chart

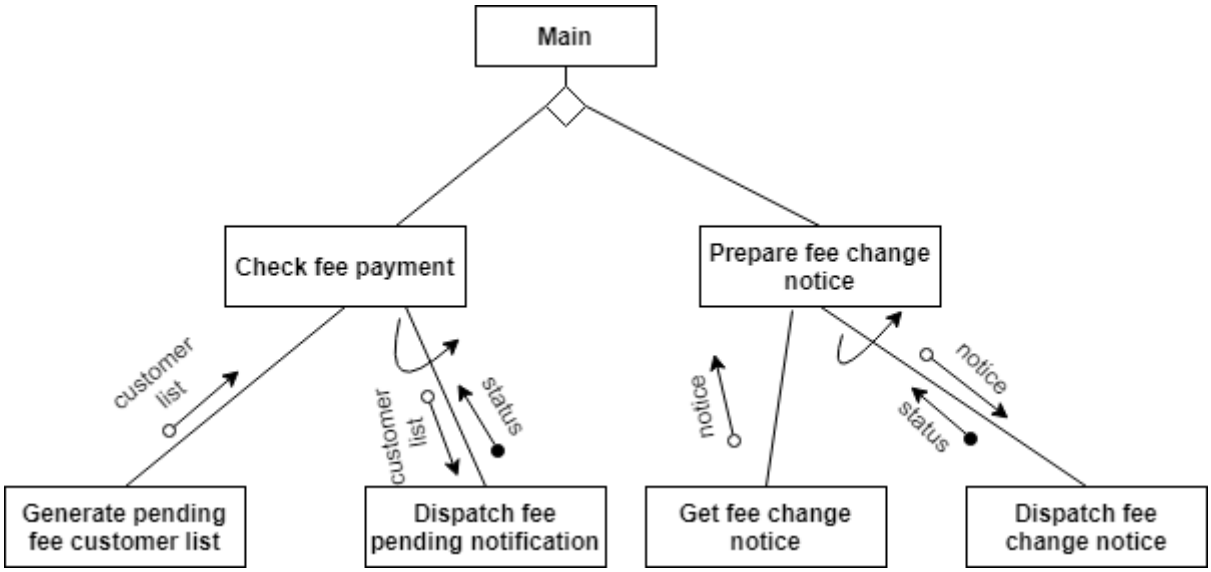

Figure 32 Structure chart for payment of a customer

3.6.3.b. Module specs

MODULE NAME: Check Fee Payment

PURPOSE: This module checks the fee payment and prepares a list of pending payments. It then notifies the customers in the list whose fee payment is pending.

PSEUDOCODE:

```
var customer_list = DB.get_customer_list()
var notice = ‘You fee payment is pending.’
var customer_id = customer_list[‘customer_id’]
var date = ‘2018/12/28’
NOTIFICATION.add_notification(notice, customer_id, date)
```

INPUT PARAMETERS : customer_list
OUTPUT PARAMETERS : notice, customer_id, date
GLOBAL VARIABLES : DB
LOCAL VARIABLES : notice, customer_id, date
CALLS: ***Generate pending fee customer list, Dispatch fee pending notification***
CALLED BY: ***Main***

4. Summary

We have successfully completed the design for the system named ‘Fitness Application’ which is implemented for a Fitness GYM according to the requirement of the scenario. While designing the required system we implemented the concept of DFD, Structure chart, Data dictionary, Module specification, Process modules and ER-diagram. It can be used in the real-world scenario. All the group members contributed for the completion of this work.

The group task contains environmental model specification, internal model specification, data dictionary and design specification of the system. The designs on group task are general surface layer design for the overall system which are further expanded and designed on individual tasks section. Furthermore, on individual tasks section, each group member has worked on Environmental model specification, internal model specification and design specification for individual functions.

We faced several issues on different parts of the report while working on the group. To solve the faced problems, we discussed among the group members. We conducted group meeting and tried to figure out solutions to the problems. We did research on concepts related to software engineering. Research was done through different books, websites, journals and other which helped us to complete this work. We conduced continuous meeting and progress discussion with the module lecturer.

The work would not have been completed without the contribution of the group members. While completing the group work, we learned to design the entire system by using the DFD, data dictionary, structure chart, E-R Diagram module specs and process specs which improved our understanding of the concepts taugh on Software Engineering module. The system design is completed according to the requirement of the group work but it can be updated according to the user requirement to implement in the real-world scenario.

After the completion of this project, software engineering skills like requirement analysis, project management, structured software engineering and problem solving are developed. Hence, this project is successfully competed by working in a small group, effectively managing the project and meeting the deadline.

5. References

beginnersbook, 2019. *beginnersbook.com.* [Online]
Available at: https://beginnersbook.com/2015/04/e-r-model-in-dbms/

bridging-the-gap, 2018. *bridging-the-gap.* [Online]
Available at: https://www.bridging-the-gap.com/data-dictionary/
[Accessed 1 January 2019].

Bruza, P. D. & Weide, T. P. v. d. W., 1989. *The Semantics of Data Flow Diagrams.* Nijmegen: University of Nijmegen, Department of Informatics, Faculty of Mathematics and Informatics.

Encyclopedia, 2016. *Encyclopedia.* [Online]
Available at: https://www.encyclopedia.com/computing/dictionaries-thesauruses-pictures-and-press-releases/module-specification
[Accessed 5 January 2019].

excelsoftware.com, 2019. *Structure Model - Structure Chart Diagrams for Software Design.* [Online]
Available at: http://www.excelsoftware.com/structuremodel
[Accessed 05 01 2019].

freetutes.com, 2019. *Structure Charts.* [Online]
Available at: https://www.freetutes.com/systemanalysis/sa6-structure-charts.html
[Accessed 05 01 2019].

guru99, 2019. *www.guru99.com.* [Online]
Available at: https://www.guru99.com/er-diagram-tutorial-dbms.html

Hathaway, T. & Hathaway, A., 2016. *Data Flow Diagrams - Simply Put!: Process Modeling Techniques for Requirements Elicitation and Workflow Analysis.* north charleston: CreateSpace Independent Publishing Platform.

searchdatamanagement.techtarget, 2019. *searchdatamanagement.techtarget.com.* [Online]
Available at: https://searchdatamanagement.techtarget.com/definition/entity-relationship-diagram-ERD

searchmicroservices, 2005. *searchmicroservices.* [Online]
Available at: https://searchmicroservices.techtarget.com/definition/data-dictionary
[Accessed 1 January 2019].

tutorialspoint.com, 2019. *SC Modules,* s.l.: s.n.

tutorialspoint, 2018. *tutorialspoint.* [Online]
Available at:
https://www.tutorialspoint.com/software_engineering/software_analysis_design_tools.htm
[Accessed 1 January 2019].