



slingshot college
(इस्लिङ्गटन कलेज)

CS5002NI Software Engineering

Allgemein

20% Group Coursework

AY 2022-2023

Group Name:			
SN	Student Name (Section name)	College ID	University ID
1)	Niwesh Dhital(L2C6)	np01cp4a210205	21049549
2)	Umang Keshar Lamichhane(L2C6)	np01cp4a210278	21049616
3)	Aauras Koirala(L2C6)	np01cp4a210115	21049465
4)	Kushal Bhatta(L2C6)	np01cp4a210186	21049531
5)	Pranab Dhungana(L2C6)	np01cp4a210211	21049555

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.

Table Of Contents

1) Introduction	1
1.1) Overall Description of the system	1
1.2) Aims and Objectives	2
2) Project Chart	3
3) SRS.....	6
3.1) Functional Requirements.....	6
3.2) Non-Functional Requirements	7
3.2.1) Design and Implementation Restrictions	7
3.2.2) Design and Implementation Constraints	8
3.2.3) External Interfaces Required.....	8
3.2.4) Other Non-functional Requirements.....	10
3.2.5) Goals of implementation	11
4) Detailed specification of GROUP task.....	12
4.1) Environmental model specification	12
4.1.1) Context Level Diagram.....	12
4.1.2) Data Flow Diagram (Level-1)	13
4.1.3) Data Flow Diagram (Level-2)	14
4.2) Internal model specification for the system.....	15
4.2.1)Entity Relationship Diagram (ERD)	15
4.2.2) Data Dictionary for Allgemein Transport System	17
4.2.3) Process Specification.....	19
4.3) Design Specification (Structure Chart)	25
4.4) Assignment Diary	27
4.4.1) Assumptions	27
4.4.2) Omissions/inconsistencies	28
4.4.3) Group member responsibilities, group meetings.....	29
5) Detailed specification of INDIVIDUAL task.....	32
5.1) Register Membership (Niwesh Dhital)	32
5.1.1) Environmental model specification.....	32
5.1.2) Internal Model Specification	33

5.1.3) Design Specification	35
5.2) Hiring vehicle(Kushal Bhatta)	38
5.2.1)Environmental Model Specifications	38
1.1. Internal Model Specifications	38
5.2.2) Design Specification	40
5.3) Booking a Cab (Pranab Dhungana).....	43
5.3.1) Environmental Model Specifications	43
5.3.2)Internal Model Specifications	44
5.3.3) Design Specification	46
5.4) Report Preparation (Aauras Koirala)	49
5.4.1) Environmental model specification.....	49
5.4.2)) Internal Model Specification.....	49
5.4.3) Design Specification	52
5.5) Joining the course(Umang Keshar Lamichhane).....	55
5.5.1) Environmental Model Specifications	55
5.5.2) Internal Model Specifications	55
5.5.3) Design Specification	57
Summary.....	59
References.....	60

List of Figures

Figure 1: Context Diagram for the Entire System	12
Figure 2: Level 1 DFD for entire system	13
Figure 3: Level 2 DFD for complete system	14
Figure 4: Entity Relationship diagram for complete system.....	16
Figure 5: Hierarchical structure of Structure Chart	26
Figure 6: Context Level Diagram for register membership	32
Figure 7: Data Flow Diagram Level-1 for register membership	33
Figure 8: Data Flow Diagram Level-2 for register membership	34
Figure 9: Structure Chart for Member Registration	35
Figure 10: Context Level Diagram for Hiring vehicle	38
Figure 11: Level 1 DFD for hiring vehicle	39
Figure 12: Level 2 DFD for hiring Vehicle.....	40
Figure 13: Structure chart for Hiring Vehicle	40
Figure 14: Context Level Diagram for booking a cab	43
Figure 15: Level-1 data flow diagram of cab booking.	44
Figure 16: Level-2 data flow diagram for booking a cab.....	45
Figure 17: Structure chart for booking cab	46
Figure 18: Context Level Diagram for report preparation	49
Figure 19: DFD level-1 for Report Preparation	50
Figure 20: DFD level-2 for Report Preparation	51
Figure 21: Structure Chart creation for Report Preparation.....	52
Figure 22: Context level diagram	55
Figure 23: DFD level 1	56
Figure 24: DFD level 2	56
Figure 25: Structure chart.....	57

List of Tables

Table 1: Project Charter for Allgemein Transport	5
Table 2: Functional Requirements for Allgemein Transport.	7
Table 3: Group Responsibilities	30
Table 4: Group meeting schedule in tabular form	31
Table 5: Structure chart of register membership	37
Table 6 Module specification of Hiring vehicle	42
Table 7: Module Specifications of booking cab.	48
Table 8: Module Specifications of report preparation	54
Table 9: Module specifications	58

1) Introduction

1.1) Overall Description of the system

The coursework represents the first group assignment of a five-member group, working on the coursework related to the 'Software Engineering' module. For the particular project, we were assigned the role of a developing team associated with a company namely '**Allgemein**' and we as a team were tasked with the job of application development. In order to efficiently conduct the tasks assigned; we conducted several meetings and performed several revisions along the way.

An application with the namesake '**Allgemein Transport**' was created and the five of us in unison added and designed a system primarily tasked with providing renting services. The system created was also created with customizability and in accordance with the guidelines provided to us by the client, we created a system to hire cab services nationwide, as well as heavy duty transportations like cargo trucks and bulldozers. Prior to the facilitation of renting services, the company was synonymous with the entertainment industry and was responsible with creating movies, making documentaries, and making games. Due to the uncertainty caused by the covid-19 pandemic, the company aligned itself with IEEE (Institute of Electrical and Electronics Engineering) to better facilitate the people in need and hence entered a foreign industry. The application creation was initiated by keeping in mind the primary requirement namely renting services performs tasks such as taking membership, track the status of booked cabs, register staffs and vehicles, hire a vehicle, provide detailed account of the user information to the concerned authorities, rate the experience, and also provides training services to people who want it.

The main objective of the project creation was to outline a system capable of providing renting services. We as a group collectively worked on the project by assigning each individual with a unique task which enabled us to make a system with all our collective quality

1.2) Aims and Objectives

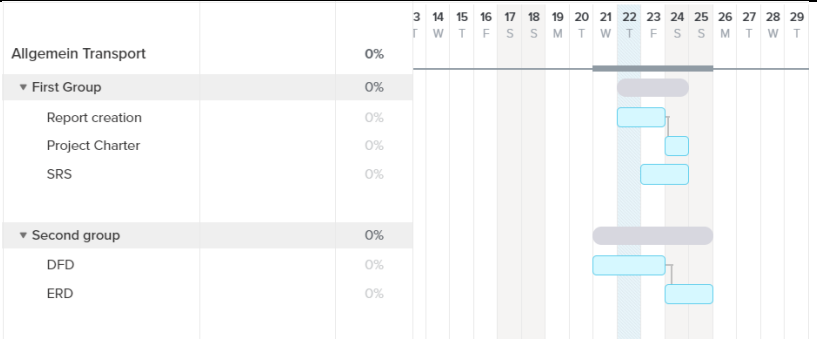
The primary aim of the project is to demonstrate adequate knowledge of 'Structured Engineering' whilst complying with pre-defined set of tasks. As a team we learned to work in unison and share ideas and opinions and mitigate unnecessary issues and band together to navigate untrodden path.

Some objectives of the project are listed below:

- To Create a booking tool for cabs or other comparable services, such as leasing large trucks from one city to the next or within a single city.
- To create a detailed report based on all the services provided by the services as mentioned above,
- After enrolling and paying a set enrolment fee that must be made public via the system, any interested individual may participate in the company's training program for operating heavy duty vehicles.
- To demonstrate practical knowledge of 'Structured Software Engineering' by staying within the constraints of a specific time period.

2) Project Chart

Project Name: Allgemein Transport	
Project Description:	To create an application capable of providing two-way services to both admin and user pertaining to services such as user information modification and manipulation as well as storage on the admin side and booking cab as well as heavy duty vehicles on the user side with an array of knicks knacks along the way.
Business Rules:	<ol style="list-style-type: none"> 1. <u>Take Membership:</u> Any individual can take membership if he/she wants to. 2. <u>Book Cab:</u> A user within the system can book a cab by specifying their destination and cab drivers can choose to accept or reject the request 3. <u>Track Driver's status:</u> The admin via the system can track the current status of the driver based on whether he/she has accepted or rejected the request. 4. <u>Staff and vehicle registration:</u> The admin via the system can choose to register and remove the details of any staff member and vehicle. 5. <u>Hire a vehicle:</u> The user embedded within the system can choose to hire any category of vehicle along with proper driving supervision. 6. <u>Report Preparation:</u>

	<p>The concerned authority connected with the system can generate reports related to customers, transaction, and business.</p> <p>7. <u>Rating the ride experience and vehicle efficiency:</u></p> <p>Customers can rate the ride experience and vehicle efficiency according to the chosen service type.</p> <p>8. <u>Join the training courses:</u></p> <p>Any interested party can join the training course provided by the company in order to manoeuvre heavy duty vehicles after enrolling and paying a certain amount of fee that should be announced via the system.</p>
<p>Timeline:</p>	 <p>The scheduled time allocation for the completion of the project is given below:</p> <p>Project layout and definition discussion: 08/12/2022</p> <p>Project initiation: 10/12/2022</p> <p>Project Control and error mitigation: 17/12/2022</p> <p>The completion pre-requisite of the project is scheduled for 5th January, 2022</p>

<p>Project is considered successful when:</p>	<p>A user is able to access the application, rent cab/s as well as heavy duty vehicles smoothly as a by-product of the system GUI being easy to use and understand.</p> <p>The admin is able to make a detailed report based on the various information being available due to the application such as the username, user email, user contact with relative ease.</p> <p>The application runs smoothly without any unnecessary bloatware bugging the application.</p>
<p>Goal Statement:</p>	<p>Create an application to help customers book a cab or a similar service also including renting heavy duty vehicles from city to city or within the city.</p> <p>The project sets out to create an application to compile step-by-step details of each of its customers and the details associated.</p> <p>The project designed meeting schedule for members to congregate and discuss issues</p>
<p>Risk:</p>	<p>However simple the project may appear, there are several flaws that are quite apparent while looked at from a technical standpoint. For example, a security breach may result in people's data being out in the public and an encryption breach may even hamper their right to privacy and though several steps were taken to make sure that the project did not face any problem, there are bound to be some errors and blunders along the way.</p>

Table 1: Project Charter for Allgemein Transport

3) SRS

3.1) Functional Requirements

Category	Functional Requirements
Take Membership	Any individual can take membership if he/she wants to.
Book Cab	<ul style="list-style-type: none">• A user within the system can book a cab by specifying their destination• Cab drivers can choose to accept or reject the request
Track Driver's status	The admin via the system can track the current status of the driver based on whether he/she has accepted or rejected the request.
Staff and vehicle registration	The admin via the system can choose to register and remove the details of any staff member and vehicle.
Hire a vehicle	The user embedded within the system can choose to hire any category of vehicle along with proper driving supervision.
Report Preparation	The concerned authority connected with the system can generate reports related to customers, transaction, and business.

Rating the ride experience and vehicle efficiency	Customers can rate the ride experience and vehicle efficiency according to the chosen service type.
Join the training courses	Any interested party can join the training course provided by the company in order to manoeuvre heavy duty vehicles after enrolling and paying a certain amount of fee that should be announced via the system.

Table 2: Functional Requirements for Allgemein Transport.

3.2) Non-Functional Requirements

3.2.1) Design and Implementation Restrictions

When developing the given application, the following design and implementation constraints should be kept in mind:

- The system will need to be scalable in order to support a large number of users, automobiles, and transactions.
- Requests and transactions should be managed by the system in real-time in order to be faster and more dependable down the line.
- The system must be able to manage simultaneous reservations for both big vehicles and taxis, thus it must have a flexible booking system that can accommodate various vehicle kinds and needs.
- Because transactions across states may involve various tax rates and local laws, the system should be able to manage such transactions.
- The system should be non-biased to mitigate review botting in order to not boost a particular driver's profile and sway some people.

3.2.2) Design and Implementation Constraints

- The **Allgemein Transport System** must manage expected and unexpected faults in a way that prevents data loss and protracted downtime.
- Each user must keep their password confidential. Each person has to have a unique fingerprint or a password only relevant and trackable to the user, stored on the concerned device (e.g., laptop, smartphone)
- Only the administrator can control user addition and deletion in the **Allgemein Transport System** and create reports.
- In the future, the system shall run across various platforms without any restriction.
- Currently, the system has not been evaluated on anything other than a Smartphone running the latest update of android. However, the system will run on any android device as well as an IOS compatible device.
- The system must use an internet connection for the user to communicate with the database.
- Users must enter their correct names and passwords to operate the system.
- The system must be created with an open-ended software to make sure that the users may down the line be able to identify and rectify bugs and errors within the system.

3.2.3) External Interfaces Required

3.2.3.1) Hardware Interfaces

- The application is compliant and workable with any smartphone device that has an online web-browser.
- HTTP/2 being prone to poor dynamic and lossy wireless connection, QUIC protocol was adopted since integrating QUIC in our apps would reduce the tail-end latencies compared to TCP.
- When the system is accessed, we may collect information pertaining to the user's device name, hardware model and other unique device identifiers in order to provide the users with a more personalized experience.

- The device running the system shall ask for permission from the user to access location services and then after track the exact location of the user to the dot and communicate the said information with the software to inevitably access the database to see the availability of drivers available for pickup.

3.2.3.2) Software Interfaces

- The system may use third parties, such as Google, to serve ads about offers and latest updates relating to the system.
- The user within the system will be able to rate and review the ride experience if he/she wishes to and when that happens, the rating given will be highlighted on the website for other registered users to view.

3.2.3.3) GUI Interface

- The System needs to use a GUI interface that is both easy to understand from the standpoint of a user as well as the customer
- The GUI must be compliant to the standards of the device on which it will function.

3.2.3.4) Communication Interface

- The system must interact with customer devices in a way that it can extract all the necessary information from the user as well as find out clever ways to work through and navigate restrictions as well.
- The system shall establish or at least try to establish a connection with the main database of the company.

3.2.4) Other Non-functional Requirements

3.2.4.1) Performance:

The system should be able to flexible enough to accommodate reparations and additions in certain sectors to accommodate heavy usage if necessary.

3.2.4.2) Security:

The system must have provisions to include extra layer of security if in case the already existing security system fails in some way.

3.2.4.3) Backup:

There should be a protocol in place to help mitigate unnecessary errors and to restore all the records within the system if in case some kind of calamity strikes, or some disaster happens

3.2.5) Goals of implementation

- The system administrator will maintain a blog to update various information to the concerned user whether it be an operator or just a regular customer about the various the upcoming and present features and offers that are relevant to the customer.
- The system shall be accessible and open to various modifications by the user and the people the user trusts with his/her login information.
- The System will open up -pathways down the line to make sure that the programmers from across the spectrum whether it be a JAVA programmer or a python programmer, both can make/ work on the same application at a time.
- Through the system, people shall be able to take special training related to driving whenever they feel like with an insignificant amount of money.
- The system shall have a feedback section solely there to listen and solve people's problems.
- The system should have a proper path for modification in order to prioritize the most significantly affecting issue to be managed quicker than a non-trivial issue that could be taken care of at a later time(e.g., updating firmware must be done quickly, synonymously whereas adding a new irrelevant or redundant feature should not be atop the priorities list.)
- Additionally, the system should have a strategy in place for periodically backing up data and restoring it in the event of an emergency.
- MySQL and Oracle are necessary to operate/maintain database.

4) Detailed specification of GROUP task

4.1) Environmental model specification

4.1.1) Context Level Diagram

4.1.1.2) Data Flow Diagram (Level-0)

A context diagram is a high-level view of a system. It is a basic sketch meant to define an entity based on its scope, boundaries, and relation to external components like stakeholders. Otherwise known as a Level 0 data flow diagram, a context diagram provides a general overview of a process, focusing on its interaction with outside elements rather than its internal sub-processes. The latter is typically reserved for more advanced data flow diagrams (PEDRIQUEZ, 2022).

The context Level Diagram of the particular project is below:

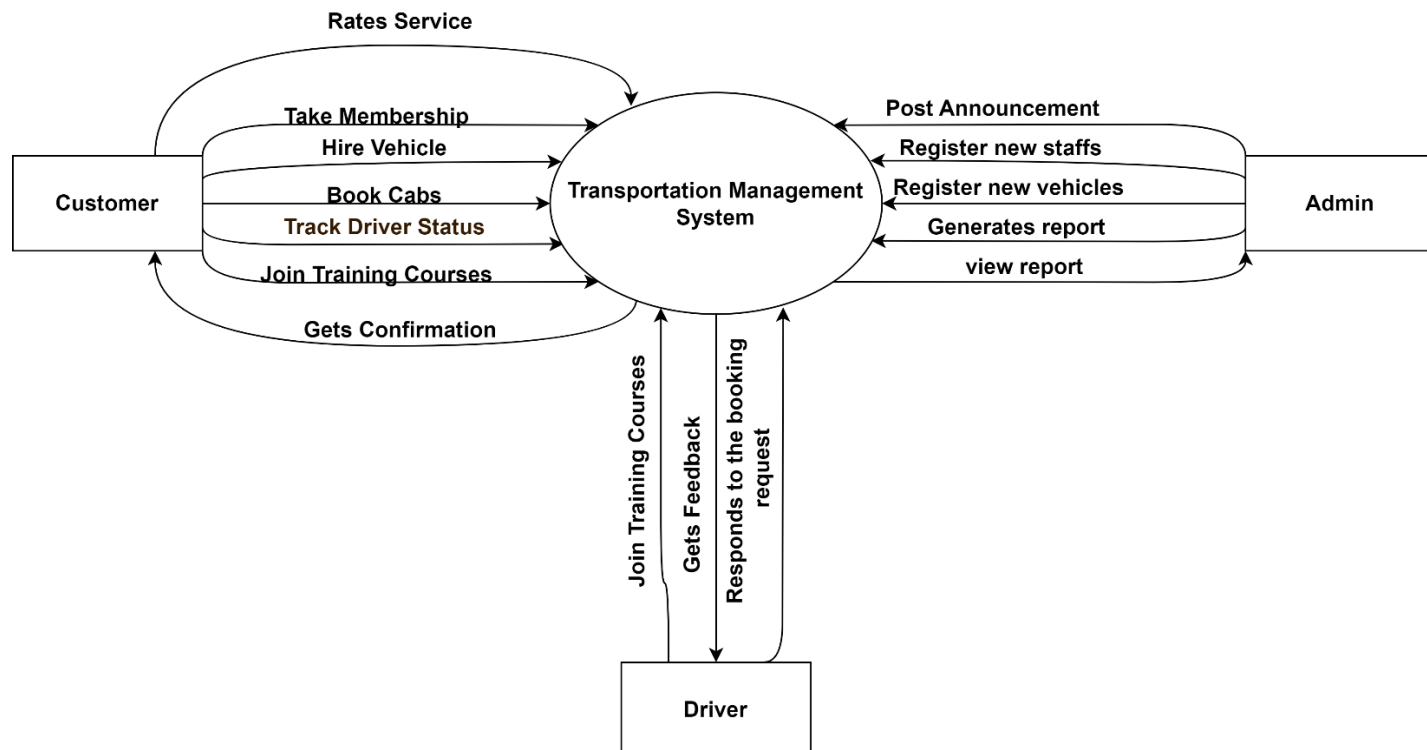


Figure 1: Context Diagram for the Entire System

4.1.2) Data Flow Diagram (Level-1)

The single process node from the context diagram is divided into sub-processes in level 1 DFD. The diagram will require more data flows and data stores when new processes are introduced to connect them. This can involve adding the room selection and enquiry processes, as well as data stores, to the reservation system in the hotel reservation example (Lucidchart, 2022). The Level-1 diagram for the particular project is displayed below:

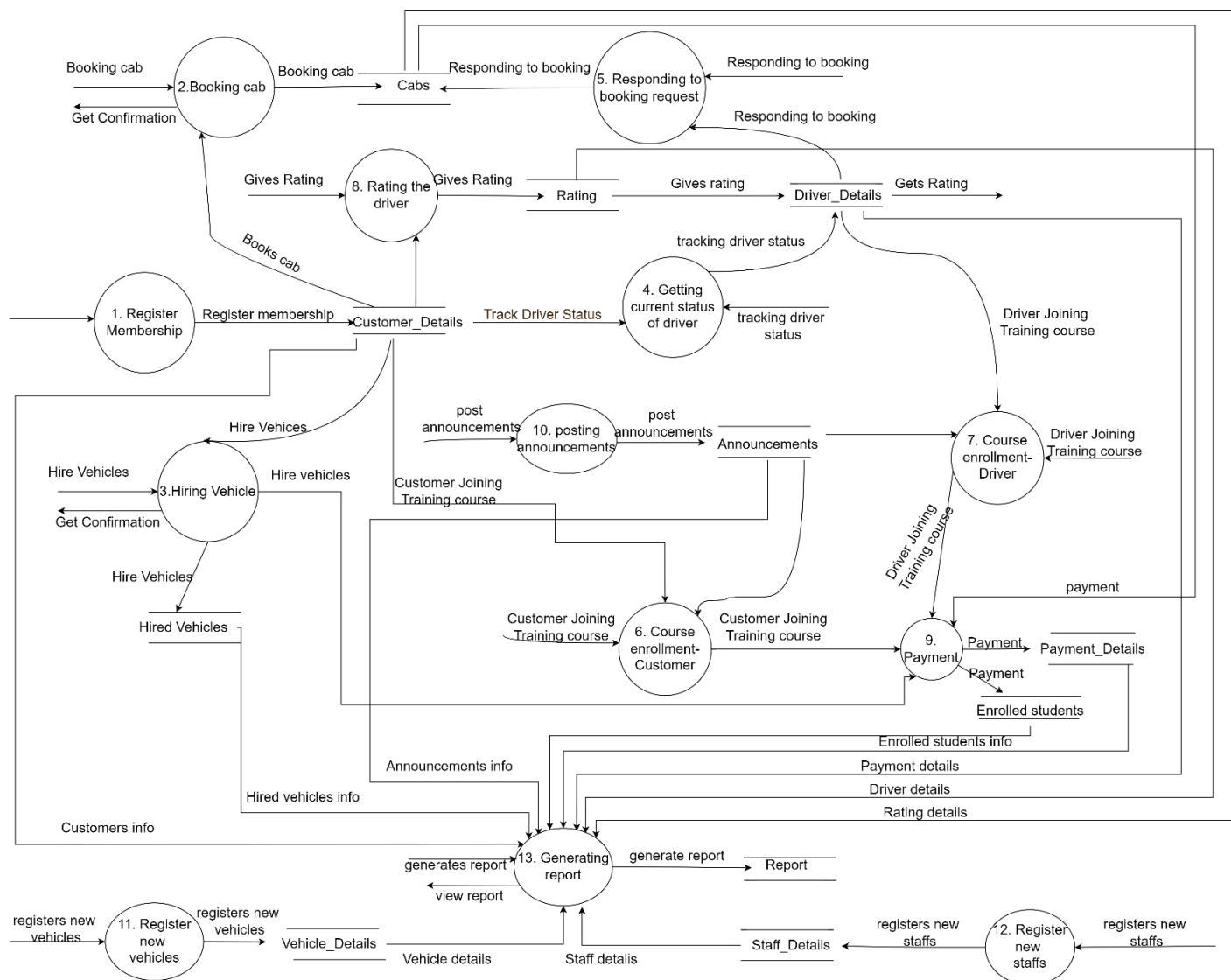


Figure 2: Level 1 DFD for entire system

4.1.3) Data Flow Diagram (Level-2)

The level-2 diagram is a further expansion on the concept of the entirety of the project that is all about creating a vehicle renting system capable of giving minute details to the admin if necessary.

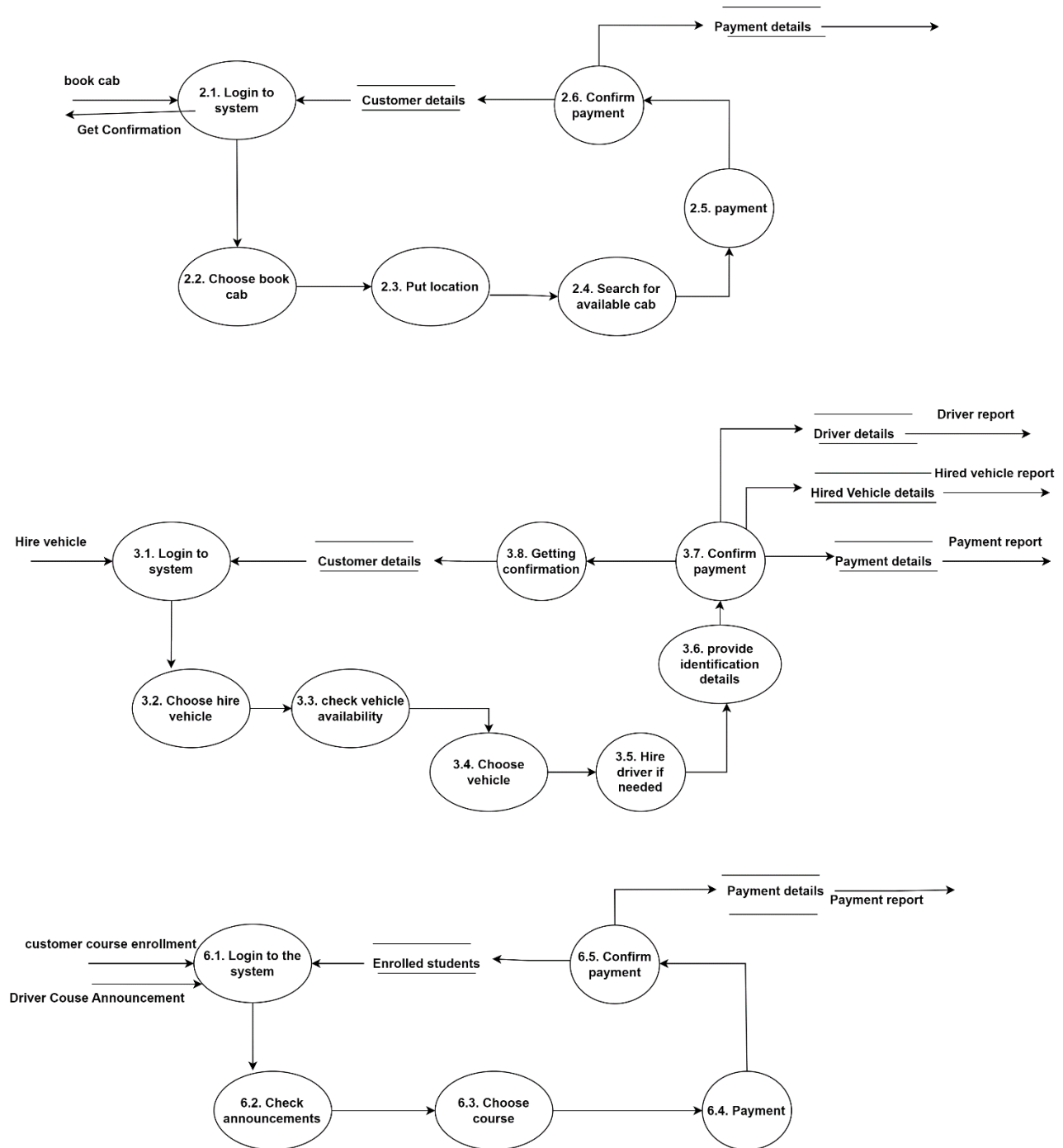


Figure 3: Level 2 DFD for complete system

4.2) Internal model specification for the system.

4.2.1) Entity Relationship Diagram (ERD)

The entity relation diagram is a visual depiction used to highlight the connections between places and objects found in a system. It gives a system a suitable beginning and end. In order to provide a standardized technique to show relationships between system components, Peter Chen first developed the ERD in 1971. The following are the main elements of an ERD.

- Entity
Entity refers to the individuals and organizations participating in the system's development. A rectangle serves as the ERD entity's representation.
- Attribute
An entity's attributes are its defining characteristics. In the ERD, it is represented by an oval.
- Relationship
Any two of the entities can be connected using this. In an ERD, it is symbolized by a diamond form.

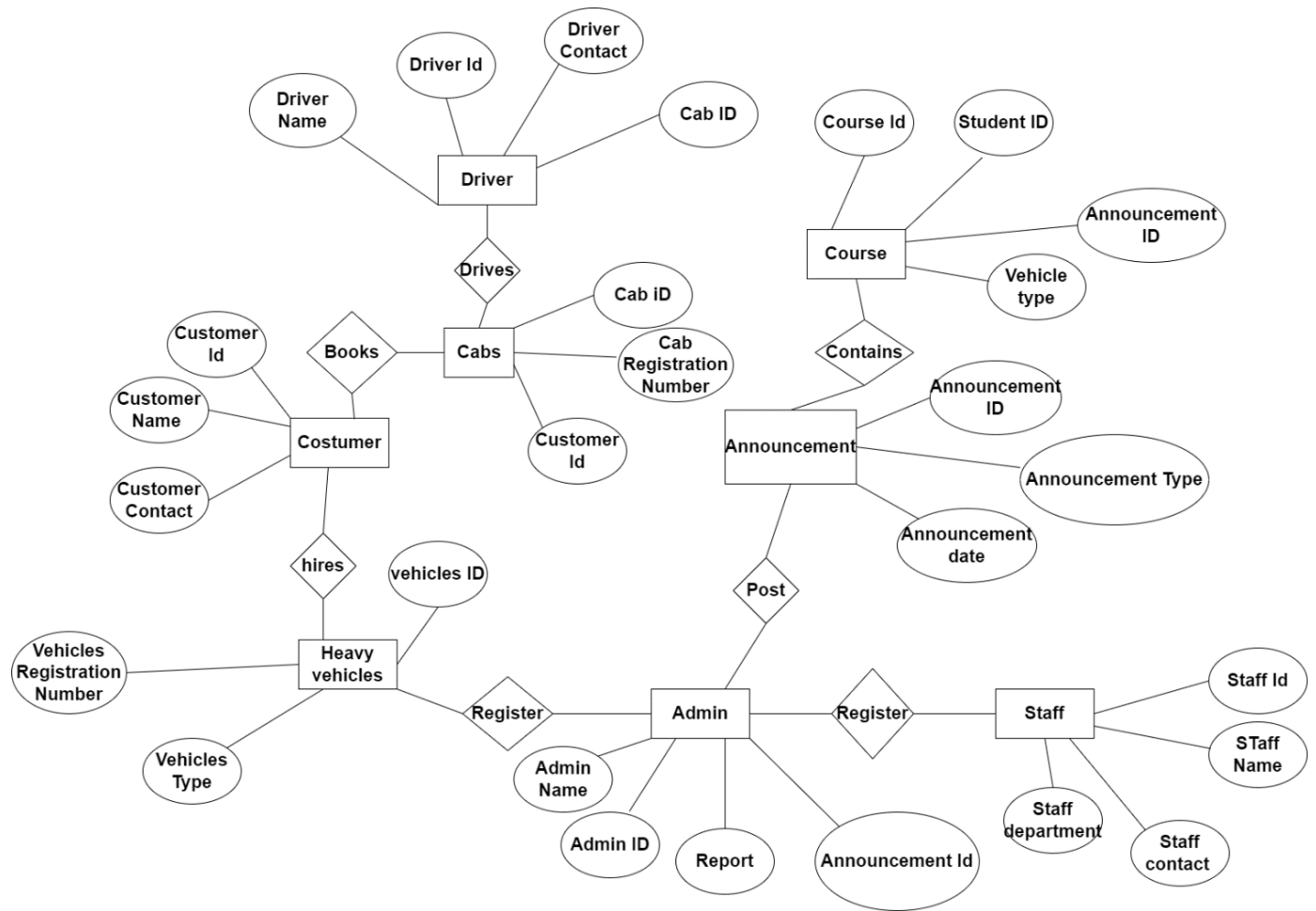


Figure 4: Entity Relationship diagram for complete system

4.2.2) Data Dictionary for Allgemein Transport System

Recruit driver = command

Register cabs = command

Register costumer = command

Register Heavy vehicles = command

Add staff as admin = command

Recruit as staff = command

Make announcement = command

Present report = command

Start the course = command

Staff = staff_Id + Staff_Name + Staff_department + Staff_Contact

Admin = Admin_ID+ Admin_Name+ Report+Announcement_id

Heavy vehicles= Vehicles_ID + Vehicles registration number+ Vehicles type

Costumer= Customer ID+ Consumer Name+ Costumer Contact

Cab= costumer Id+ Cab ID+ Cab registration number

Driver= cab id+ Driver Contact+ Driver Id+ Driver Name

Announcement = Announcement ID + Announcement Type + Announcement date

Course= Course ID+ Student ID + Announcement ID+ Vehicle type

Staff info={recruit fresh staff + Staff}*

Admin info = {New admin + Admin}*

Heavy vehicles info = {new heavy vehicle + heavy vehicles}*

Costumer info = {Enrol costumer + Costumer}*

Cab info = {newly registered cab + cab}*

Driver info={new driver + driver}*

Announcement info={new announcement + Announcement}*

Course info = {new course + course}*

4.2.3) Process Specification

A process specification is a method used to document, analyse, and explain the decision-making logic and formulas used to create output data from process input data. Its objective is to flow down and specify regulatory/engineering requirements and procedures. High-quality, consistent data requires clear and complete process specifications (Techopedia, 2012).

Process Number: 1

Process Name: Register membership

Description: This process lets new user to take a membership.

Input Data Flow: Take membership

Output Data Flow: Register membership

Detailed Logic: The logic for this process is as follows:

- Record the details of members in a customer details database.
- Get the details of new user to take membership.

Process Number: 2

Process Name: Booking cab

Description: This process lets the customer to book a cab.

Input Data Flow: Booking cab

Output Data Flow: Get confirmation + Booking cab

Detailed Logic: The logic for this process is as follows:

- Allow the customers to book a cab.
- Sends confirmation to the users.
- Store the booked cab in cabs database.

Process Number: 3

Process Name: Hiring vehicle

Description: This process lets the customer to hire vehicles.

Input Data Flow: Hire vehicles

Output Data Flow: Get confirmation + Hire vehicles

Detailed Logic: The logic for this process is as follows:

- Allow the customers to hire vehicles
- Sends confirmation to the users.
- Store the hired vehicles into hired vehicles database.

Process Number: 5

Process Name: Responding to booking

Description: This process lets the driver to respond to the booking done by customer.

Input Data Flow: Responding to booking

Output Data Flow: Responding to booking

Detailed Logic: The logic for this process is as follows:

- Gets driver details from driver details database to respond to the booking.
- Update the available cabs in Cabs database.

Process Number: 6

Process Name: Course enrolment -customer

Description: This process lets the customer to join for training course.

Input Data Flow: customer joining training course

Output Data Flow: Customer joining training course

Detailed Logic: The logic for this process is as follows:

- Gets the customer details from customer details database.
- Gets announcements details from Announcements database.

Process Number: 7

Process Name: Course enrolment -Driver

Description: This process lets the driver to join for training course.

Input Data Flow: Driver joining training course

Output Data Flow: Driver joining training course

Detailed Logic: The logic for this process is as follows:

- Gets the driver details from driver details database.
- Gets announcements details from Announcements database.

Process Number: 8

Process Name: Rating the driver

Description: This process lets the customer to rate the driver from whom they got the service.

Input Data Flow: Gives rating

Output Data Flow: Gives rating + gets rating

Detailed Logic: The logic for this process is as follows:

- Get customer details from customer details database.
- Stores the rating details in Rating database and updates rating of driver in driver details database.

Process Number: 9

Process Name: Payment

Description: This process lets the customer to pay for the service used.

Input Data Flow: Payment + customer joining training course + driver joining training course + hire vehicle

Output Data Flow: Payment

Detailed Logic: The logic for this process is as follows:

- It lets the user to make a payment of services they have used.
- Stores the payment details in payment details database.
- Stores the customer who had pay for training course in Enrolled students database.

Process Number: 11

Process Name: Register new vehicles

Description: This process lets the admin to register new vehicles.

Input Data Flow: Register new vehicles

Output Data Flow: Register new vehicles

Detailed Logic: The logic for this process is as follows:

- Get the details of new vehicles to register.
- Record the details of vehicle in a vehicle details database.

Process Number: 13

Process Name: Generating report

Description: This process prepares or generate the reports of payments, hired vehicles, drivers, announcements, customers, rating, etc.

Input Data Flow: generates report + Customers info + Hired vehicles info+ Announcements info + Enrolled students' info + Payment details + Driver details + Rating details + vehicle details + staff details

Output Data Flow: Generate report + view report

Detailed Logic: The logic for this process is as follows:

- Get customer info from mock customer details database.
- Get hired vehicles info from hired vehicles database.
- Get announcements info from announcements database.
- Get Enrolled students' info from enrolled students database.
- Get Payment details from payment details database.
- Get driver details from driver details database.
- Get rating details from Rating database.
- Get vehicle details from vehicle details database.
- Get staff details from staff details database.
- Filter the details obtained and prepare a report.

Process Number: 2.1

Process Name: Login to system

Description: This process let members to login to the system.

Input Data Flow: Book cab + Member details

Output Data Flow: Get confirmation + Choose service

Detailed Logic: The logic for this process is as follows:

- Gets customer details from customer details database.

4.3) Design Specification (Structure Chart)

A structure chart is a tool for software design that depicts the relationship between processing components. It explains the transmission of data between the modules' hierarchy of components. It includes transaction analysis and input-to-output transformation analysis. In computer software, structure charts display the relationships between processing components. It is a design tool that shows the connections between program modules graphically. It displays how each module in a system interacts with one another and graphically represents the data that are passed back and forth. Before writing any software code, structure charts are created. They pinpoint the data transfers that take place among several modules that communicate with one another (freetutes.com, n.d.).

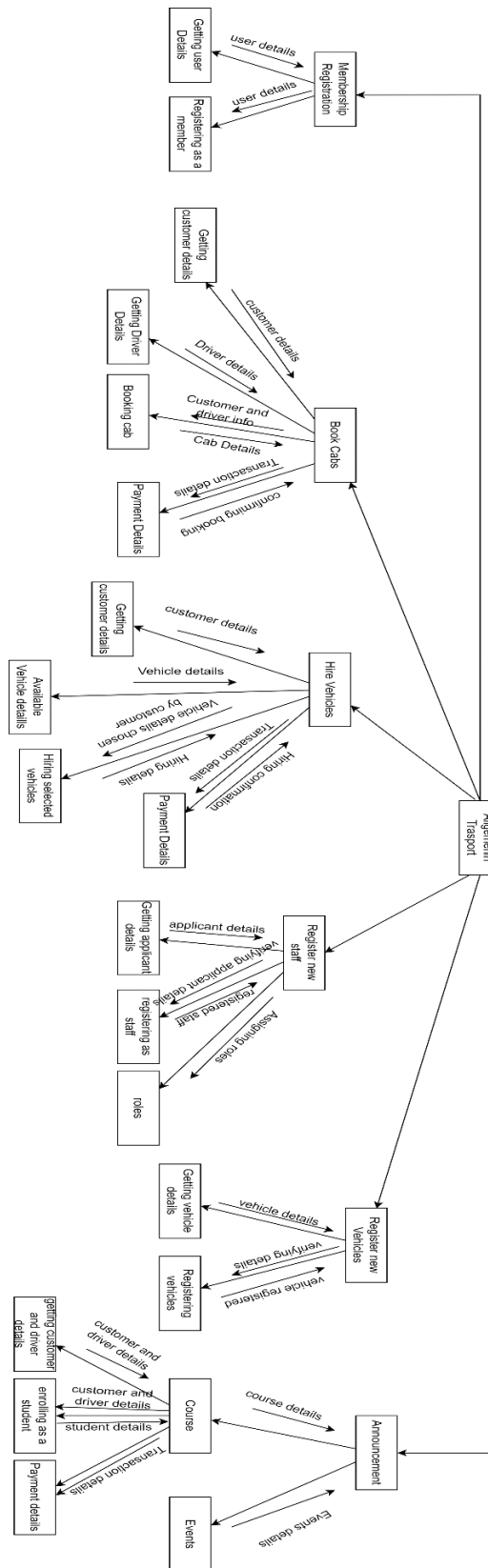


Figure 5: Hierarchical structure of Structure Chart

4.4) Assignment Diary

The coursework was assigned to us on 22nd November,2022 and since then we banded together to meet the deadline of 5th January,2022. During the initiation phase, we held several discussion sessions, and we assigned tasks pertaining to the strength of each individual.

4.4.1) Assumptions

During the creation of DFD or Data Flow Diagram for the particular project, there were several assumptions that were there for consideration. However, in addition to that, the five of us banded together to produce assumptions to make our tasks easier and to flesh out the project even more. The few assumptions we made are as follows:

- Customers can hire drivers as well as vehicles as the two are separate entities within the system.
- Customers are required to pay a certain undisclosed amount of money upfront even before consuming or using any of the multitude of services.
- However, even before proceeding with payment, the customers must verify their identification by providing an ID.
- Drivers and new customers can unroll for the training course with their being no discrepancy of any kind in between.
- There is a provision where the customers can pay only after the driver accepts their requires of a cab and the driver maintains the payment.
- Admin has the privilege to register staff and vehicles and no other entity can do the same.
- An admin has the privilege of creating and maintaining the report with the admin having the final say in the type of information to be included in the report as well.
- Drivers are stored in 'Driver' database all throughout the system.
- All the drivers within the system are denoted by '**Driver**' entity all throughout the system and every single driver is grouped into the 'Driver' entity with a congruent database name 'Driver' as well.

- Customers are recognized within the system by 'Customer' entity and all customer records are stored in 'Customer' database.

4.4.2) Omissions/inconsistencies

The project in its entirety was given to us in a very through format. However, there are bound to be some inconsistencies in the project regardless of the level of the final product and some that we found are as follows:

- **Pressing Deadline**

The project as we initiated was going to be a small and very less time-consuming ordeal. However, as we carried forward with all our tasks, it turned out to not be the case and a project of such complexity we believe needs a more lenient timeframe.

- **Un-specified use of resources**

When we were called upon to make the project, the instructions listed no limitations as to the use of resources and in real life such an inconsistency could lead to increase in cost and irritation on both the customer as well as the developer's side.

- **Incomplete operational rules**

When we were assigned the project, we were not provided with enough business/operational rules to the point that the team had to make sever assumptions.

- **Unclear budged**

In order to make a business successful, the project owner must be aware of the amount of money to use and distribute. However, that was not the case for the particular project

4.4.3) Group member responsibilities, group meetings.**4.4.3.1) Group responsibilities**

The detailed account of task allotment is given below:

Members Name	Designated tasks
Niwesh Dhital	<ul style="list-style-type: none">• Contributed to managing and management aspects of the project namely assigning tasks to each individual• Contributed to the analysis portion of the project creation by trying to mitigate as much flaws as possible.• Contributed to SRS creation and completion• Co-created project charter for the project.• Co-wrote introduction and conclusion portion of the project.• Contributed to creating and compiling report for the overall project completion.
Umang Keshar Lamichhane	<ul style="list-style-type: none">• Contributed to SRS creation and completion• Co-created project charter for the project.• Co-wrote introduction and conclusion portion of the project.• Kept a record of weekly meetings and solved

	<p>problems associated with the project.</p> <ul style="list-style-type: none"> • Fleshed out the Graphical User Interface Portion of the project by creating a visual representation.
Kushal Bhatta	<ul style="list-style-type: none"> • Co-created DFD (Data Flow Diagram) • Co-wrote pseudocode for the project • Co-created Entity Relationship Diagram (ERD) and data-dictionary for the project
Auras Koirala	<ul style="list-style-type: none"> • Created the module specification portion • Co-created the introduction portion of the project
Pranab Dhungana	<ul style="list-style-type: none"> • Co-created DFD (Data Flow Diagram) • Co-wrote pseudocode for the project • Co-created Entity Relationship Diagram (ERD) and data-dictionary for the project

Table 3: Group Responsibilities

4.4.3.2) Group Meetings

Date of meeting	Task of the day	Achievements
2022-12-20	Task Distribution	Tasks to all the members based on their personal advocacy.
2022-12-21	Report Preparation and DFD Creation	Developing things like introduction and Definitions, Data flow diagram
2022-12-22	SRS Creation, DFD completion	Completion of DFD and SRS developed inside SRS Functional requirements.
2022-12-23	SRS completion	SRS completion, namely Non-functional requirements.
2022-12-25	Project Charter and ERD creation and completion	Project charter and ERD completion.
2022-12-26	Structure chart and individual task	Started the creation of Structured chart and individual task.
2022-12-27	Structure chart and individual task	Structure chart and individual task completion.
2022-12-28	Individual task	Individual task completion.
2022-12-29	Data dictionary	Worked on data dictionary

Table 4: Group meeting schedule in tabular form

5) Detailed specification of INDIVIDUAL task

5.1) Register Membership (Niwesh Dhital)

The particular project centric application includes a feature to help people register as members. Members are required to provide information such as their name and ID that the system verifies and sends over to the main database to register the member information.

5.1.1) Environmental model specification

5.1.1.1) Context Level Diagram

The particular context level diagram was made with the intention of highlighting the membership process used by new members wanting to be embedded in the system.

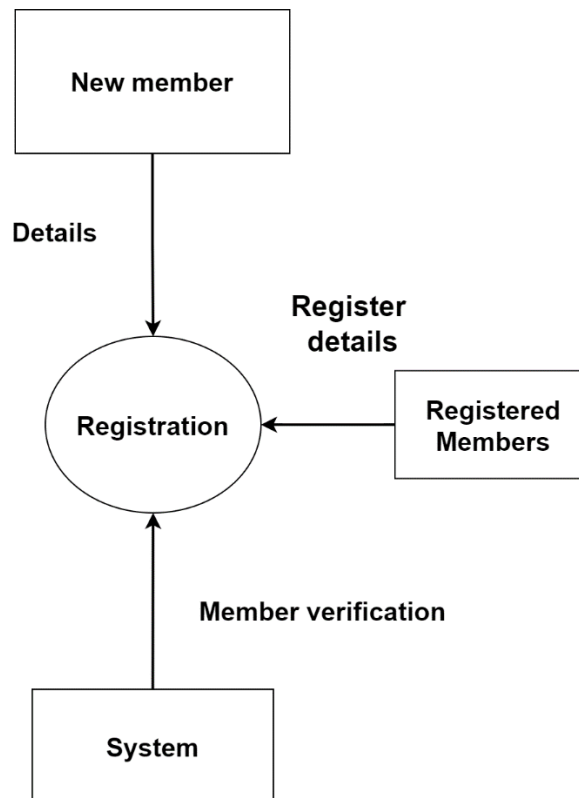


Figure 6: Context Level Diagram for register membership

5.1.2) Internal Model Specification

5.1.2.1) Level-1 Data Flow Diagram(DFD)

The level 1 DFD was designed to highlight the membership registration process and once data are validated across the entire diagram with multiple processes of data validation and once member details are validated to create registered member entity, the end processed data is then sent over to create a valid member Identity card in form of Members ID

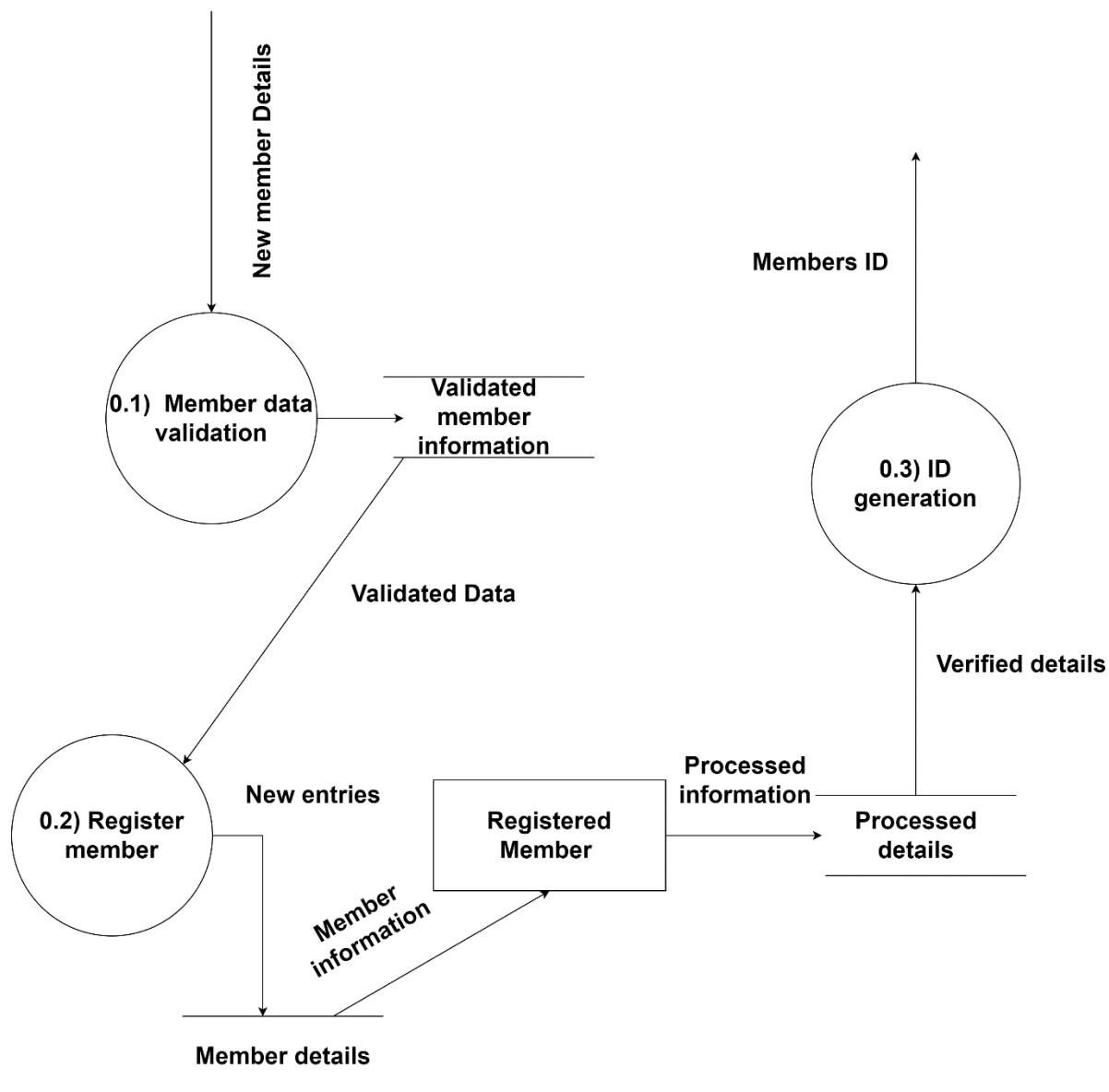


Figure 7: Data Flow Diagram Level-1 for register membership

5.1.2.2) Level-2 Data Flow Diagram(DFD)

The particular Level-2 diagram was created to highlight the details of becoming a member in even greater detail. The technique was specifically intended to reverify the registration standard and to update the already existing registered members list and now for a second time an ID is created for the user with the only difference here being that the information this time is open to the public to see. It also includes various standards in the form of processes to validate the records within the system. The particular iteration of DFD creation includes the several additional processes all aiding to verify the information with the system.

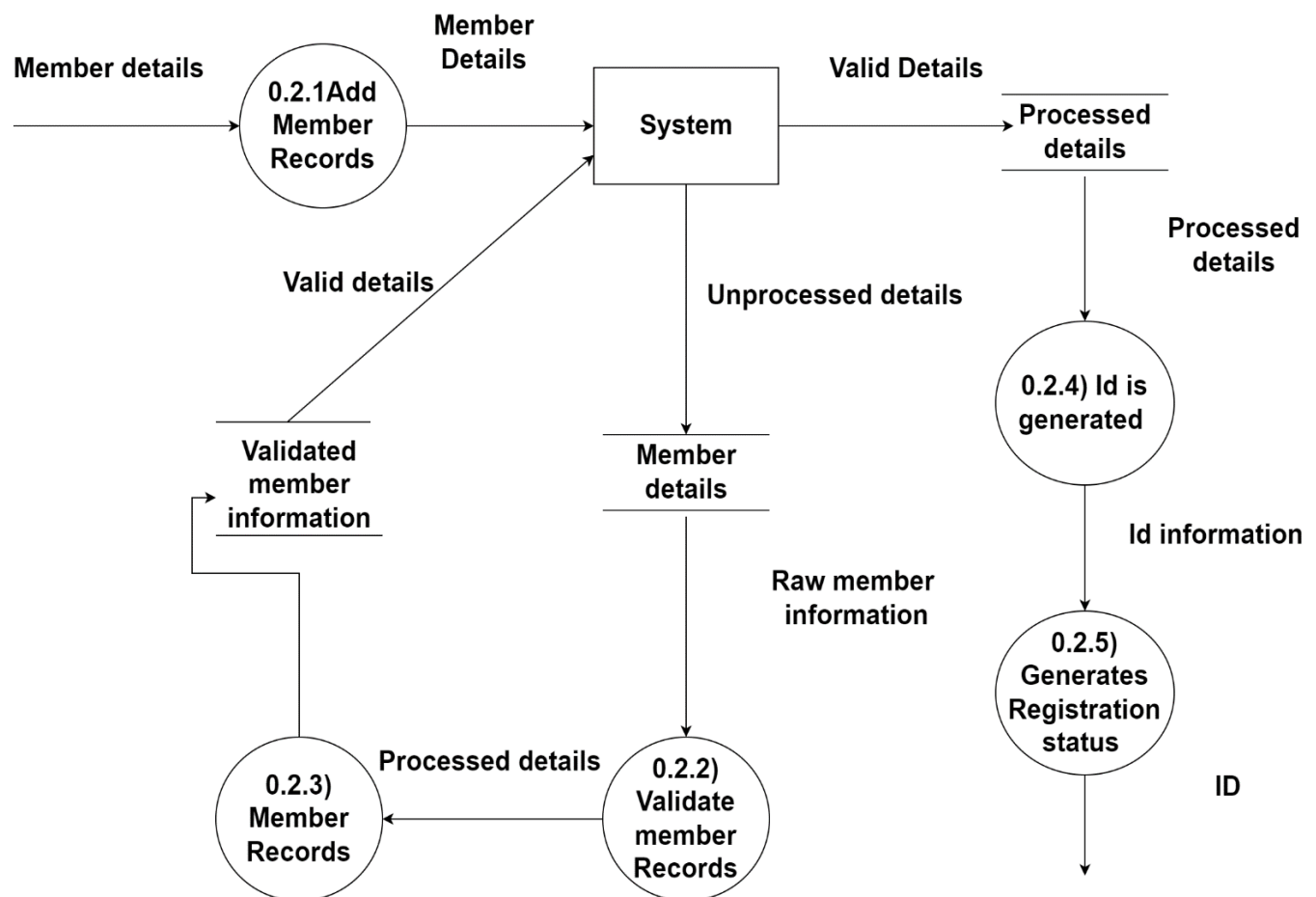


Figure 8: Data Flow Diagram Level-2 for register membership

5.1.3) Design Specification

5.1.3.1) Structure Chart

This diagram shows the overall layout of the Register a member application. This diagram shows the steps involved in registering a member such as adding a member, verifying the member information, validating the member, and checking the registration status of the member. The add member and verification process are the two fields or parameters of the structure chart. Verification process is a step to verify member information and the validation parameter also includes a field of its own where it checks duplication/ field requirement. After the completion of the former step, members are validated within the system and a registration status is displayed indicating the end of the process.

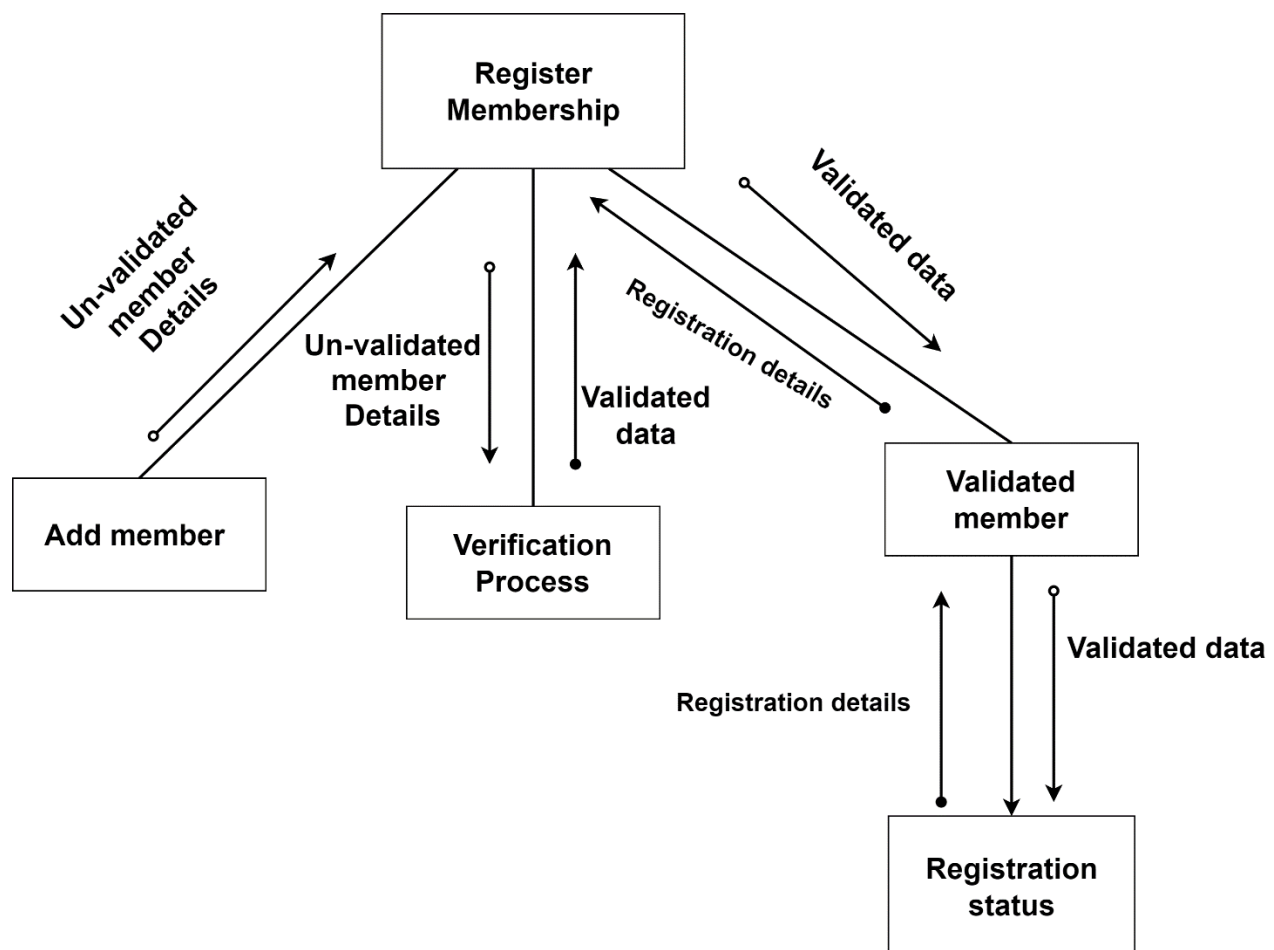


Figure 9: Structure Chart for Member Registration

5.1.3.2) Module Specification

Name	Register Membership
Purpose	To Register Membership
Pseudocode	<pre> DO INPUT member_name INPUT Member_ID INPUT Registration_status DO IF member_name, Member_ID, Registration_status information is verified END DO DISPLAY("The member Id:" + Member_ID) DISPLAY Registered Member END DO </pre>
Input Parameters	Member details
Output Parameters	Identification status(member ID)

Global Variable	None
Local Variable	Member ID
Call	Registration status
Called By	member

Table 5: Structure chart of register membership

5.2) Hiring vehicle(Kushal Bhatta)

This vehicle rental management system allows customer to hire several types of construction vehicles and specialist driver(if needed). In this system customer can hire vehicles by providing their identification documents and provide their location where they want to deliver the vehicle.

5.2.1)Environmental Model Specifications

5.2.1.1) Context Level Diagram (Level 0)

The process of renting a car is represented in the context level diagram with the information given by the client and the admin's validation of the information.

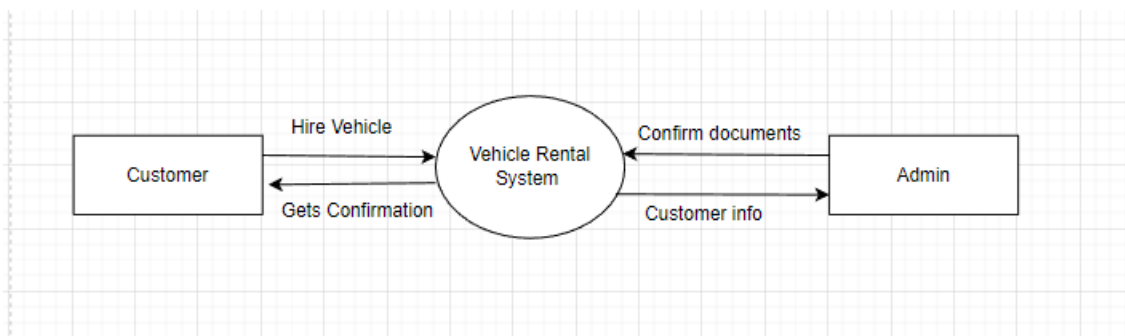


Figure 30: Context Level Diagram for Hiring vehicle

1.1. Internal Model Specifications

5.2.1.2) Level 1 DFD

This system's Level 1 data flow diagram describes the process of renting a vehicle by a client, storing the vehicle in a database, and receiving a report on the hired vehicle. The hired driver was also updated in a database and received a report on the hired driver.

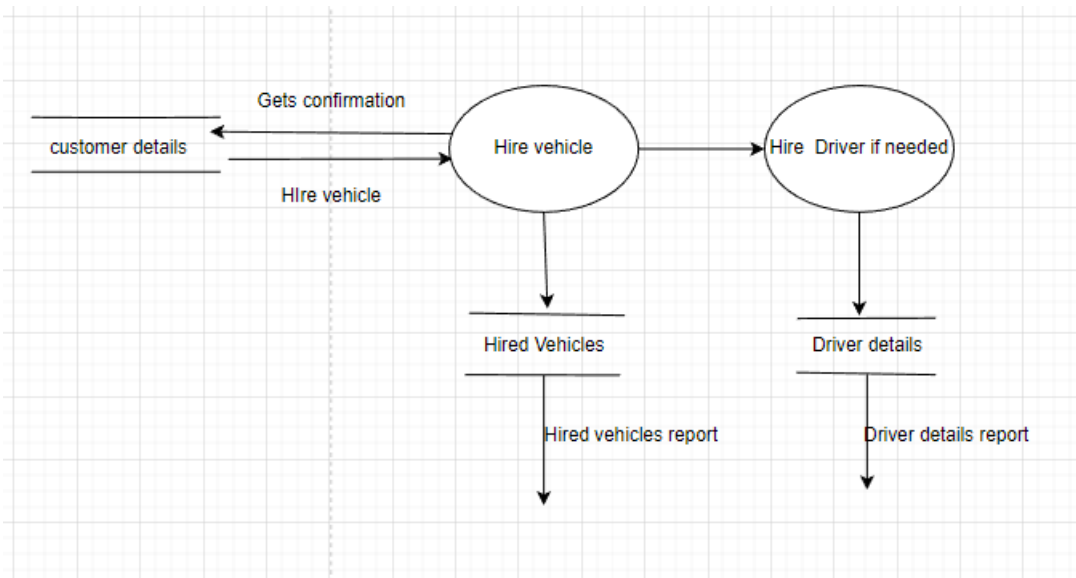


Figure 11: Level 1 DFD for hiring vehicle

5.2.1.3) Level 2 DFD

This level 2 data flow diagram contains precise information on how vehicles are hired in the system. All of the processes are shown below in the diagram, such as how the consumer receives confirmation of the hired vehicle and where the data of hired vehicle and hired driver is stored.

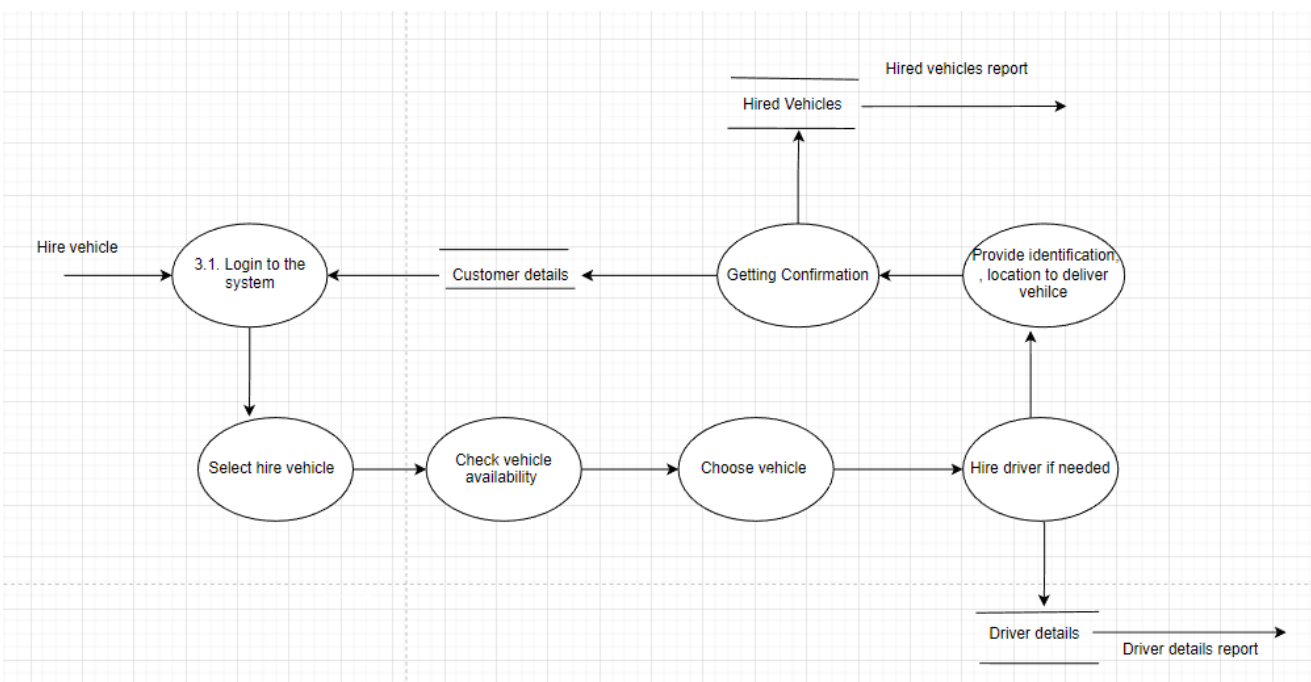


Figure 12: Level 2 DFD for hiring Vehicle

5.2.2) Design Specification

5.2.2.1) Structure Chart

The following structure chart depicts the modules utilized for vehicle hiring:

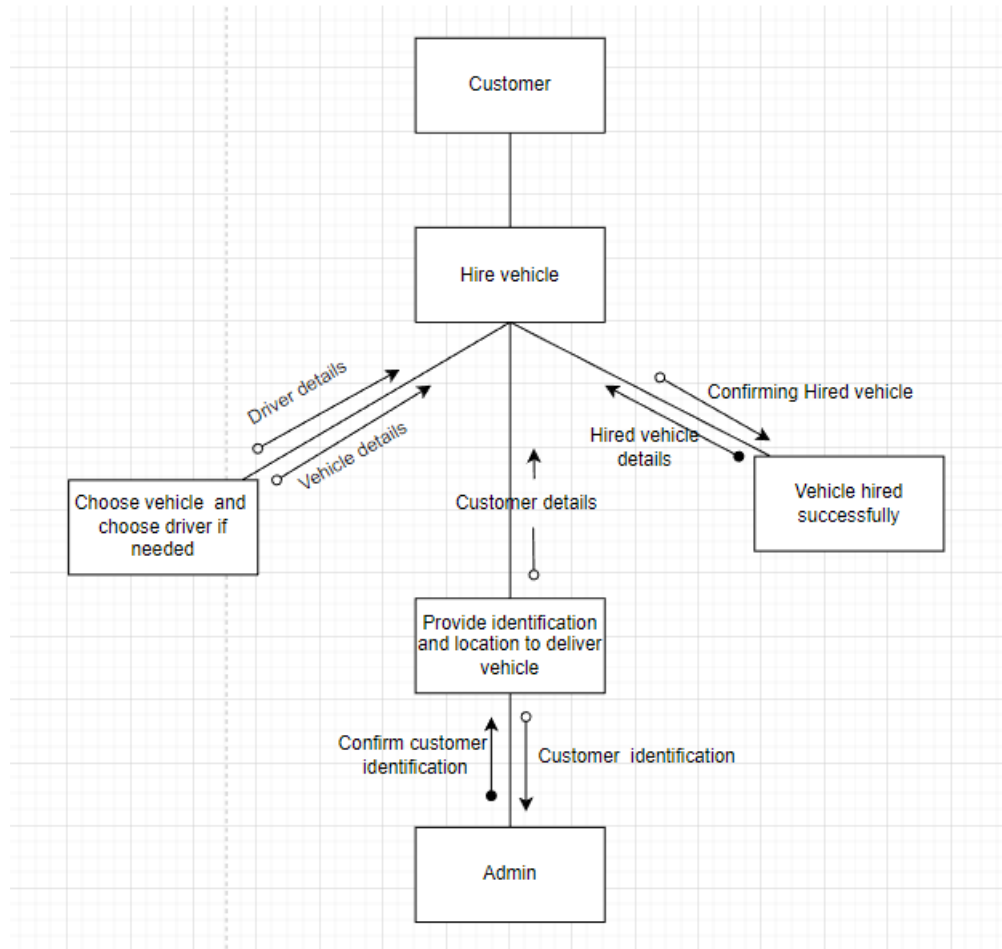


Figure 13: Structure chart for Hiring Vehicle

5.2.2.2) Module Specifications:

Name	Hire Vehicle
Purpose	To hire vehicle
Pseudo code	<pre> DO INPUT customer_Contact INPUT customer_Identification INPUT delivery_Location DO IF vehicle_availability OR driver_availability is true IF admin_confirmation is true THEN provide vehicle_details ELSE Display "Vehicle not available. Please check another vehicle" END DO END DO END DO </pre>
Input Parameters	Customer Contact, customer identification, vehicle delivery location
Output Parameters	vehicle_details
Global Variables	

Local Variables	customer_Contact, customer_Identification, delivery_Location, admin_confirmation
Calls	Get admin_confirmation, display admin response
Called By	customer

Table 6 Module specification of Hiring vehicle

5.3) Booking a Cab (Pranab Dhungana)

This transportation management system has the feature which allows users to book the cab by providing their mandatory details like their ID and location. Then the system forward it to the nearby driver and driver responds to the cab request after which customer gets the confirmation of the request from the system getting the details provided from driver.

5.3.1) Environmental Model Specifications

5.3.1.1) Context Level Diagram

The context level diagram shows the process of booking cab using the details provided by customer and response of the driver to the request done by customer.

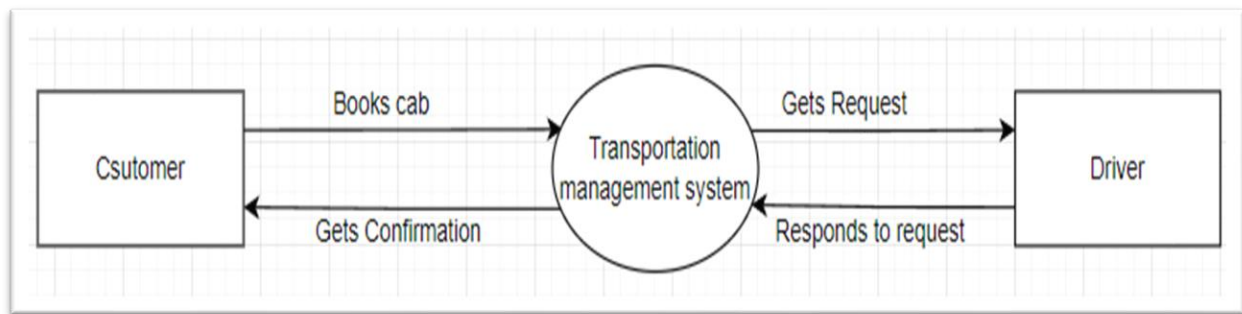


Figure 14: Context Level Diagram for booking a cab

5.3.2) Internal Model Specifications

5.3.2.1) Level-1 DFD

Level-1 data flow diagram for this system describe the process of booking the cab getting the details of customer and driver and then storing those combined data as cab details and getting the cab report as output from cab details.

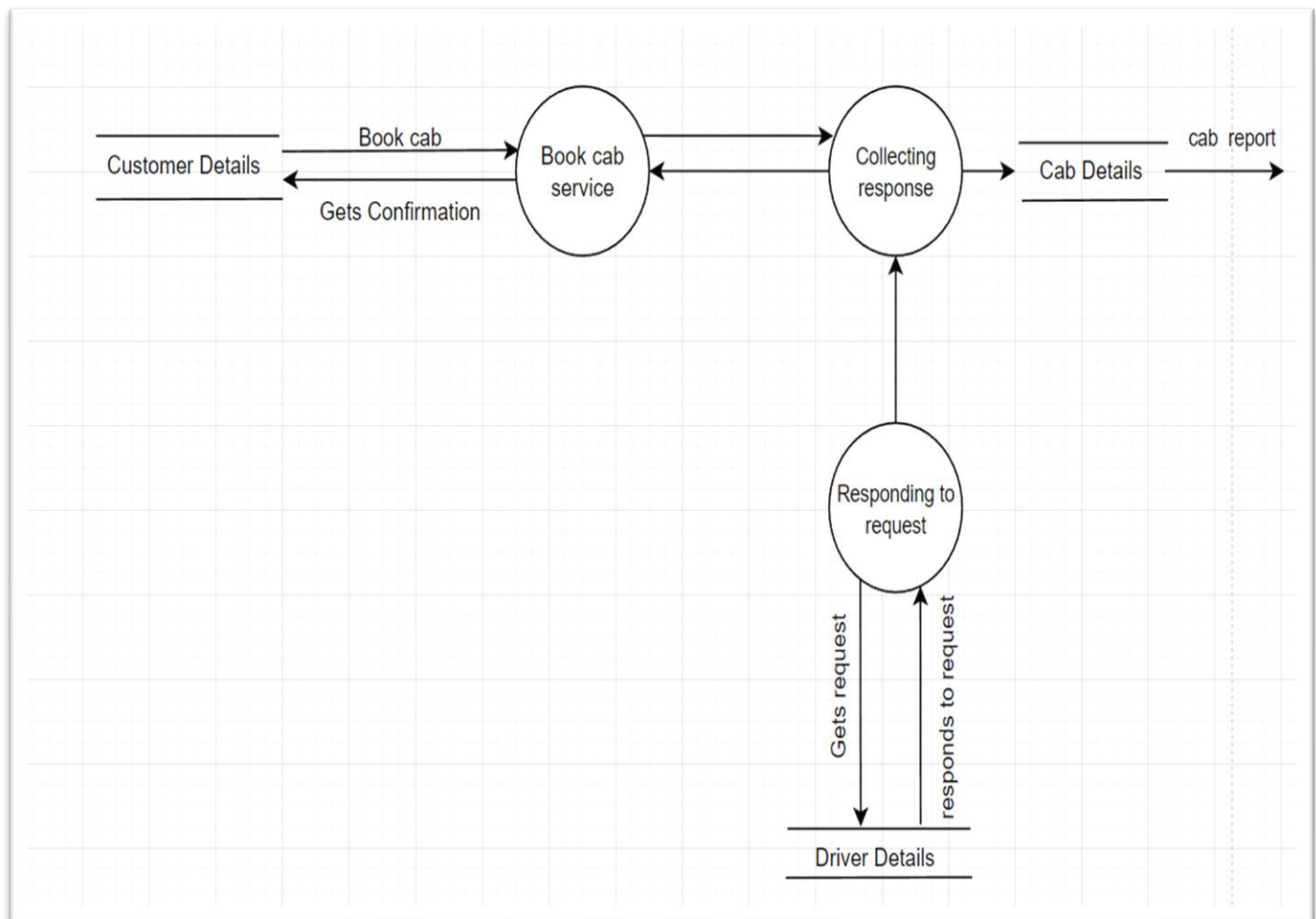


Figure 15: Level-1 data flow diagram of cab booking.

5.3.2.2) Level-2 DFD

In this level of data flow diagram, there is a detailed information about how function “Booking Cab” work in actual system. Here all the process are mentioned and shown how the data is being stored and how are driver and customer are communicating with each other at the time of booking and also how the customer and driver details are working together to make new data storage named “Cab details” which is used to see the details of currently booked cab.

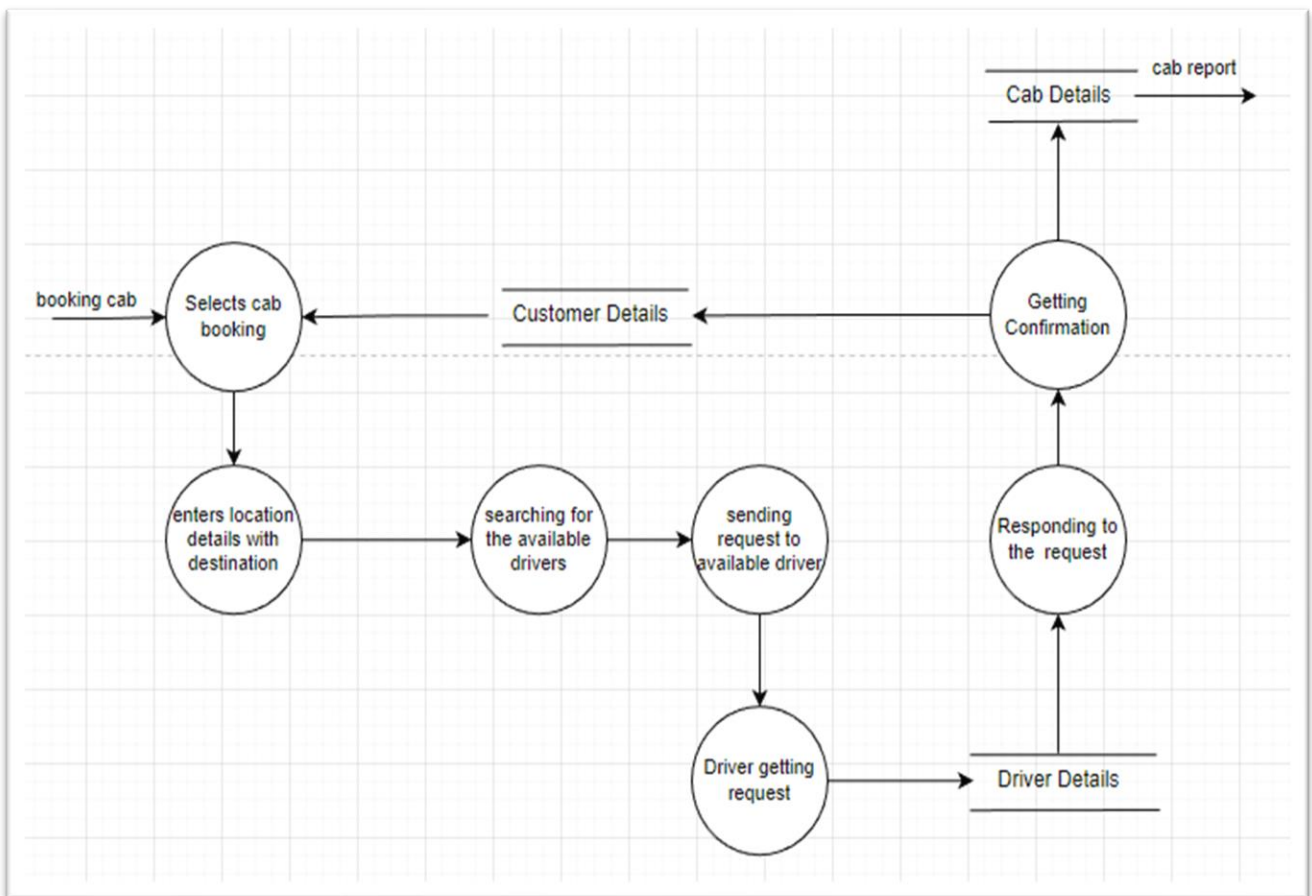


Figure 16: Level-2 data flow diagram for booking a cab

5.3.3) Design Specification

5.3.3.1) Structure Chart

Structure chart showing the modules used for booking cab is represented below:

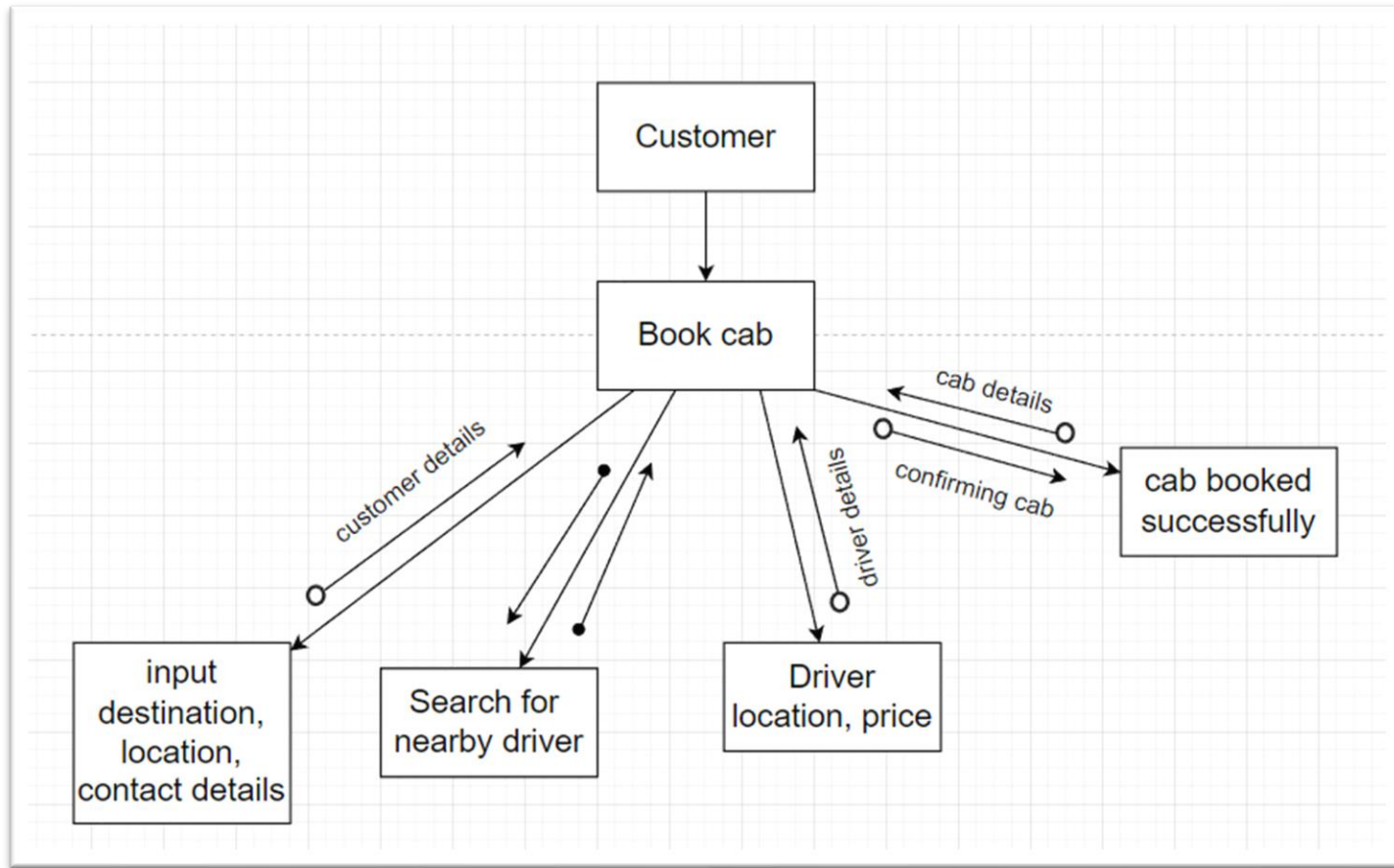


Figure 17: Structure chart for booking cab

5.3.3.2) Module Specifications

Name	Book Cab
Purpose	This module is for establishing a proper communication between the customer and cab driver. This module helps the customer and drivers to exchange their details in very convenient way and helps to book the cab online taking their details.
Pseudo code	<pre>DO INPUT cust_Email INPUT cust_location INPUT cust_destination DO IF driver_response is true THEN generate confirmation ticket, cab_details ELSE Display "No cab available." END DO END DO END DO</pre>

Input Parameters	Customer Location, Email, Destination
Output Parameters	Confirmation ticket, cab_details, driver response
Global Variables	allgemeinSys
Local Variables	cust_Email,cust_location,cust_destination,driver_response
Calls	Get driver response, display driver response
Called by	customer

Table 7: Module Specifications of booking cab.

5.4) Report Preparation (Aauras Koirala)

5.4.1) Environmental model specification

5.4.1.1) Context Level Diagram

The context level diagram was created with the goal of emphasising report preparation following system activities involving the customer, driver, and vehicle.

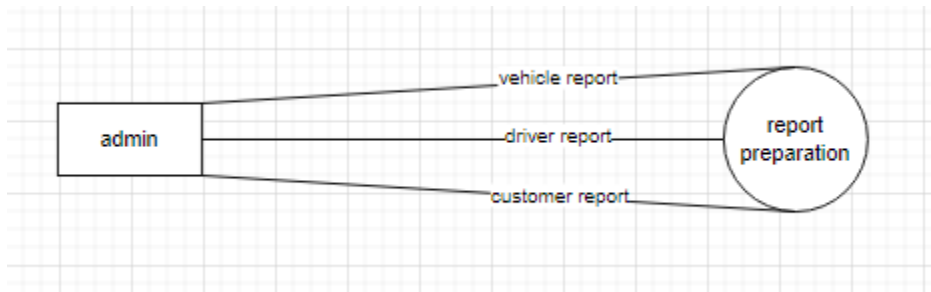


Figure 18: Context Level Diagram for report preparation

5.4.2)) Internal Model Specification

5.4.2.1) Level-1 Data Flow Diagram(DFD)

The level 1 DFD was created to show the report production process; following the system's retrieval and admin evaluation of the driver, vehicle, and customer data and the final report is created.

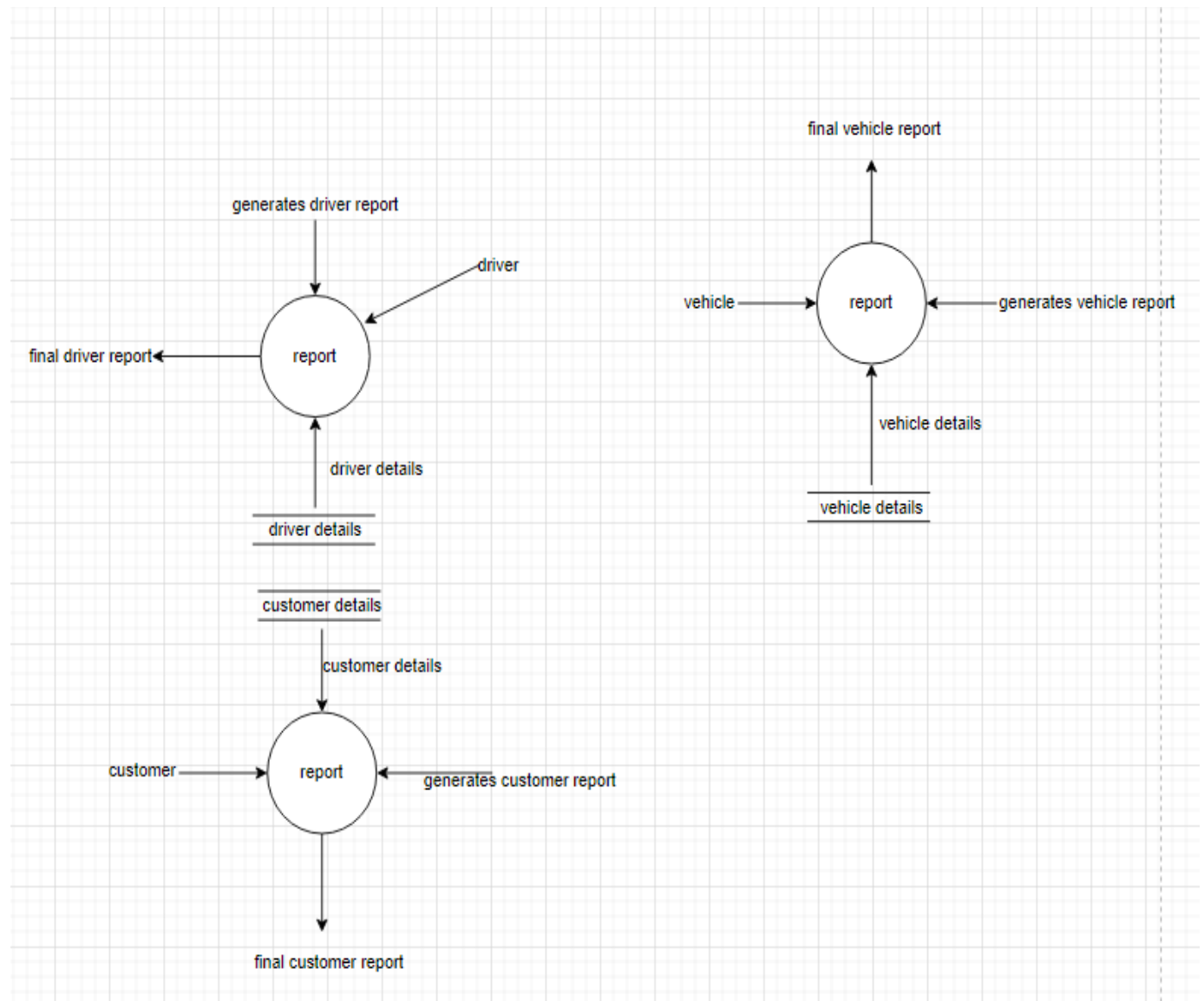


Figure 19: DFD level-1 for Report Preparation

5.4.2.2) Level-2 Data Flow Diagram(DFD)

A DFD's Level 2 is developed to display the process in greater detail. Here, customer, vehicle, and driver information are first obtained, then it is merged to create a report, and finally, when a display report command is given, the report is displayed.

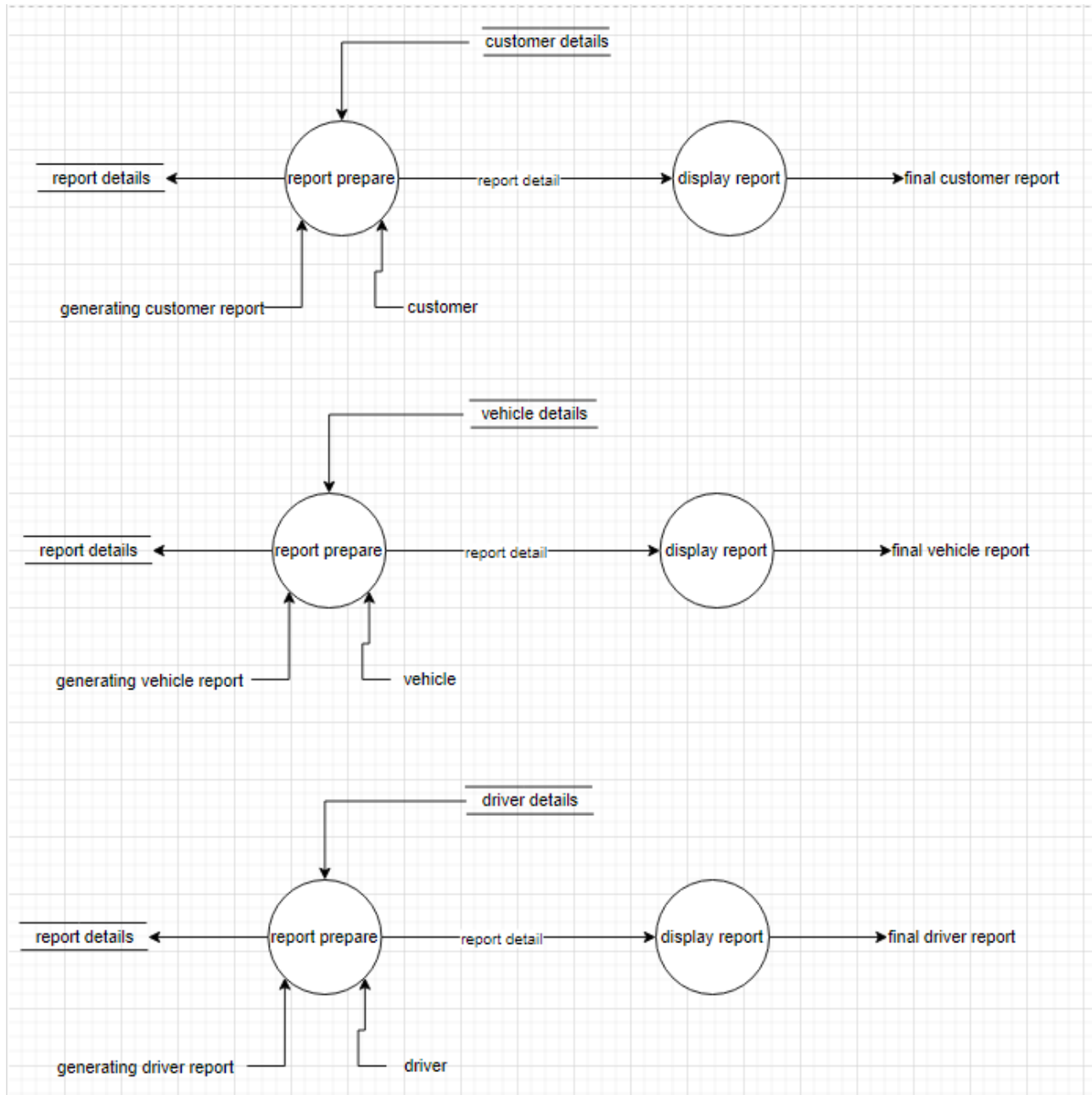


Figure 20: DFD level-2 for Report Preparation

5.4.3) Design Specification

5.4.3.1) Structure Chart

The following structure diagram illustrates the modules utilized in report preparation:

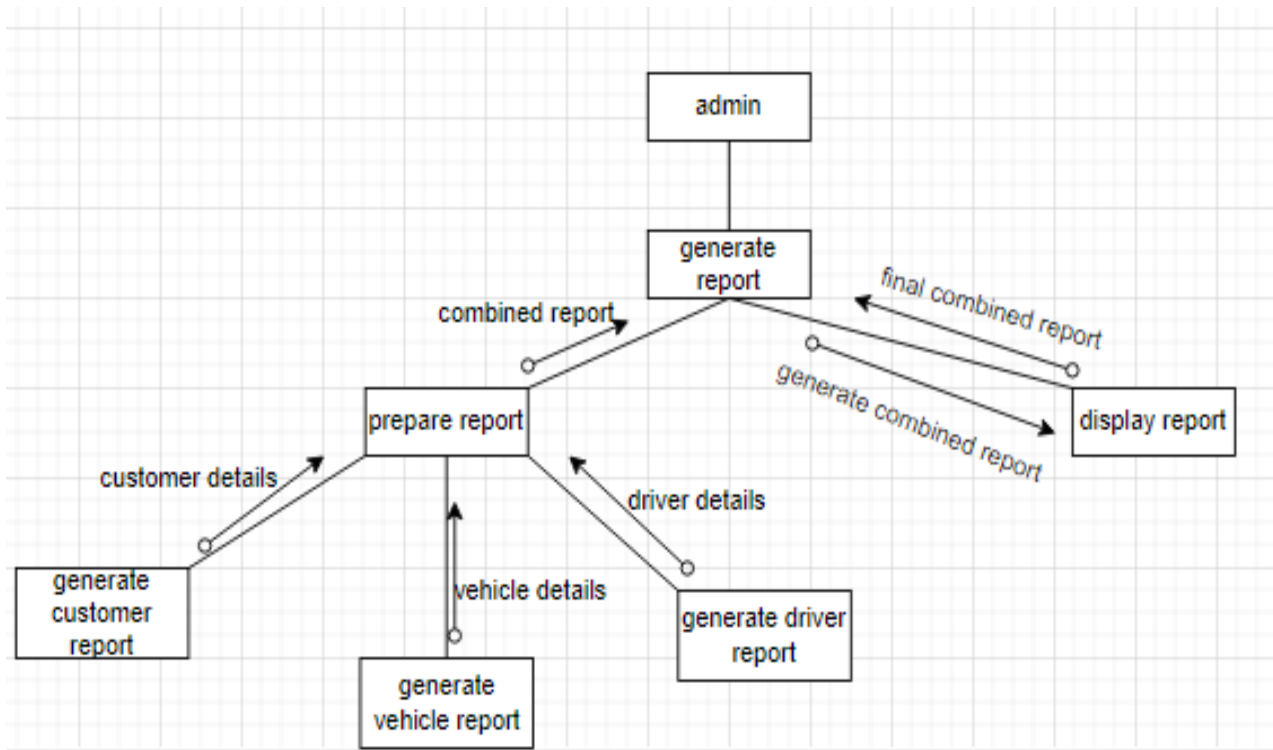


Figure 21: Structure Chart creation for Report Preparation

5.4.3.2) Module Specification

Name	Report Preparation
Purpose	To prepare a report
Pseudocode	<pre> DO var customer_details=AB.get_customer_details() var vehicle_details=AB.get_vehicle_details() var driver_details =AB.get_driver_details() var customer_id = customer_details['customer_id'] var driver_id = driver_details['driver_id'] var vehicle_id = vehicle_details['vehicle_id'] var report = generate_report(customer_details, vehicle_details, driver_details) Display(report) </pre>

	END DO
Input parameter	customer_id, vehicle_id, driver_id
Output parameter	Customer report, vehicle report, driver report
Global variable	AB
Local variable	Customer_id, vehicle_id, driver_id, report
Call	get_Customer_details, get_vehicle_details, get_driver_details
Called by	admin

Table 8: Module Specifications of report preparation

5.5) Joining the course(Umang Keshar Lamichhane)

This transportation management system has the feature which allows users to book a course for proper training to be able to manage any kind of vehicles that this company provides as a service. The system first takes user details and then books that person to an open seat in the training course after the payment, the system registers that seat to that person.

5.5.1) Environmental Model Specifications

5.5.1.1) Context level diagram

The context level diagram shows the process of booking a sit for the course using the details provided by customer and respond as conforming the seat after the payment's done.

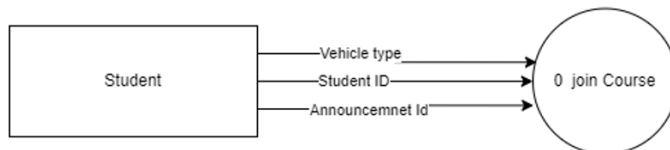


Figure 22: Context level diagram

5.5.2) Internal Model Specifications

5.5.2.1) Level-1 DFD

Level-1 data flow diagram for this system describe the process of booking seat getting the details of customer vehicles type they want to learn and the announcement id from where they got the knowledge about the course.

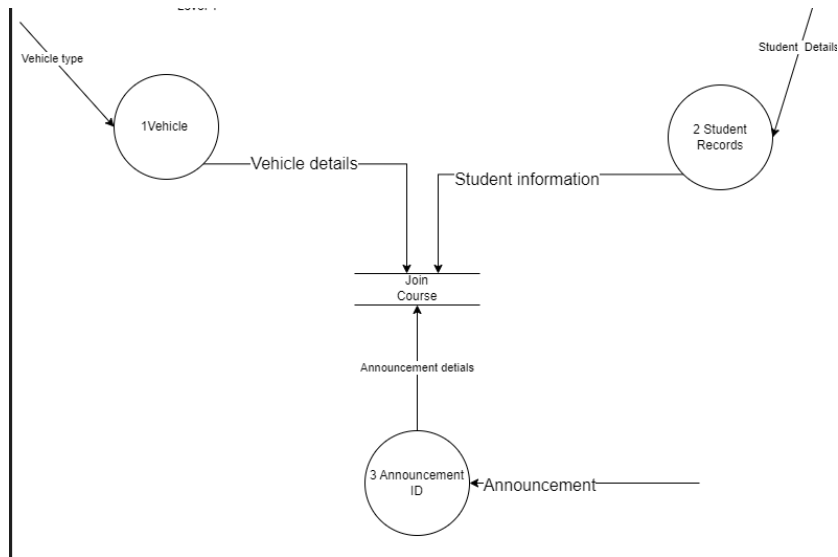


Figure 43: DFD level 1

5.5.2.2) Level-2 DFD

The level 2 was to show how the costumer can enroll in the course in a detailed manner. Here a costumer's details were first gathered and then payment is to be made then the admin verifies the payment then the user can join the course.

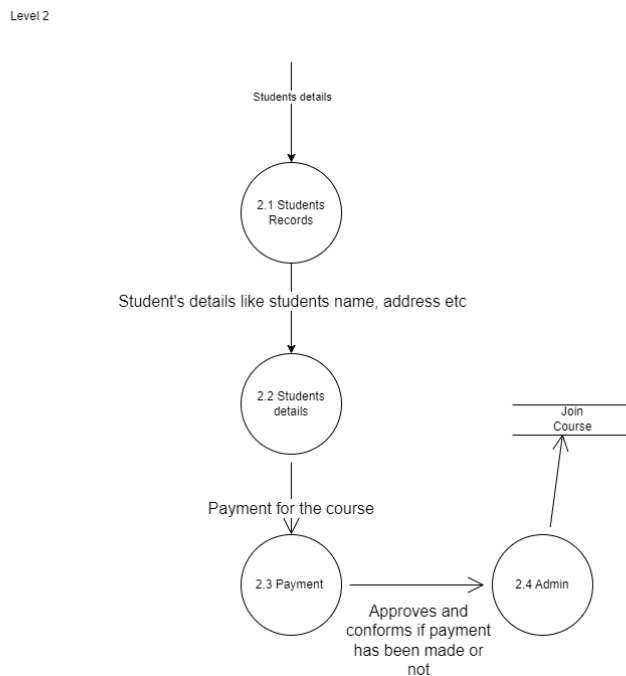


Figure 24: DFD level 2

5.5.3) Design Specification

5.5.3.1) Structure Chart

Structure chart showing the modules used for Joining the course is represented below:

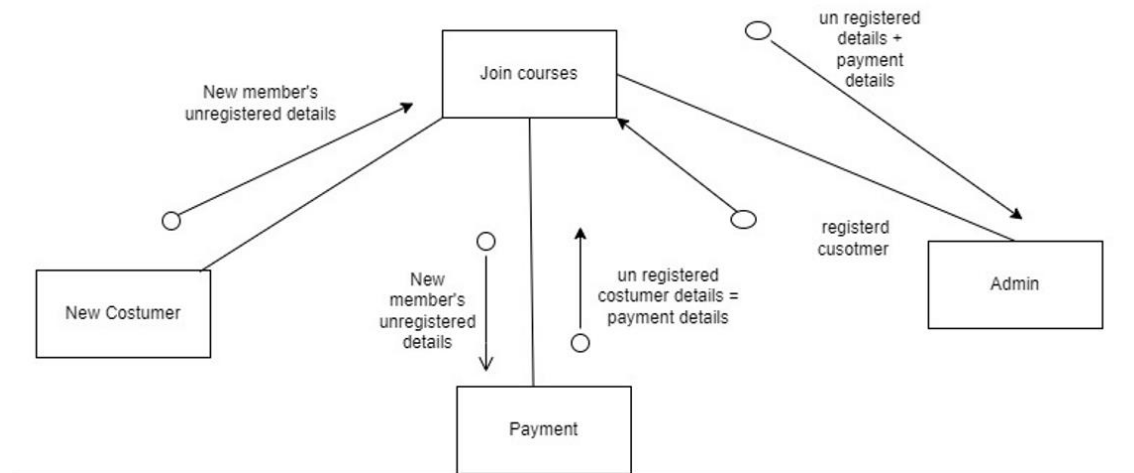


Figure 25: Structure chart

5.5.3.2) Module Specifications

Name	Join the course
Purpose	To join the training
Pseudocode	<pre> DO INPUT user_name INPUT payment INPUT Course_Id DO IF user_name and Payment are verified in the system END DO DISPLAY ("you can join the course" +Course_id) DISPLAY ("Course joined successfully") END DO </pre>
Input parameter	User details Payment details
Output parameter	Course_id
Global variable	None
Local variable	Course_id
call	Join
Called by	Join the training

Table 9: Module specifications

Summary

The purpose of the coursework was to aid 'Allgemein Transport,' a transportation company to make an application to primarily perform tasks pertaining to registration of customers and delivery of services with certain business rules in place to dictate the flow of the entire project and the allotment of tasks to certain sectors.

The project was started out by allotting various tasks to all five members within the group by specifically catering to their demands after creating a comprehensive diagram highlighting all our individual tasks such as to register members, enrol staff, join training courses, book a vehicle, and book a cab. After that, the team slowly started creating a concept level diagram to see the flow of the system and after the particular endeavour, the team swiftly moved on to create an Internal Model Specification Diagram also known as DFD or Data Flow Diagram. There were several two distinguishable levels of DFD's that we needed to create and so we did. We started out with creating level-1 DFD and swiftly moved on to create level-2 with a few hiccups here and there. After the creation of Data Flow Diagram, the team collectively banded together to create an ER-diagram where there was supposed to be an accumulation of all attributes of all the listed entities. The team finished the task with no problem. During the entirety of the project creation, the team had to create a comprehensive report that we all did after the completion of specifically listed tasks as mentioned above. Overall, the coursework was completed with no problems and only a few problems here and there.

Overall, the project was successful due to the team's gruelling and un-wavering confidence and also because of the unrelenting support we got from all our peers as well as the tutors.

References

freetutes.com, n.d. *Structure Charts*. [Online]

Available at: <https://www.freetutes.com/systemanalysis/sa6-structure-charts.html>

[Accessed 30 December 2022].

geeksforgeeks.org, 2019. *Software Engineering | Structure Charts*. [Online]

Available at: <https://www.geeksforgeeks.org/software-engineering-structure-charts/>

[Accessed 30 December 2022].

Lucidchart, 2022. *Data flow diagram examples, symbols, types, and tips*. [Online]

Available at: <https://www.lucidchart.com/blog/data-flow-diagram-tutorial#:~:text=Level%20DFDs%20are%20still,stores%20to%20link%20them%20together.>

[Accessed 31 December 2022].

PEDRIQUEZ, D., 2022. *What is a Context Diagram (and How Can You Create One)?*.

[Online]

Available at: <https://venngage.com/blog/context-diagram/>

[Accessed 31 December 2022].

Techopedia, 2012. *Process Specification*. [Online]

Available at: <https://www.techopedia.com/definition/28636/process-specification#what-does-process-specification-mean>

[Accessed 31 January 2022].