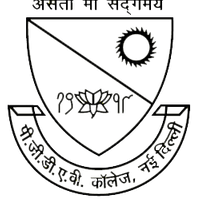
**P.G.D.A.V. College (M), University Of Delhi**

****

**B. Sc. (H) Computer Science (II Year)**

**SEMESTER IV**

**Software Engineering Project**

**SARAL**

**SUBMITTED TO: SUBMITTED BY:**

**Dr. Aparna Datt Hardik Bhanot**

**20053570013(3913)**

**Kuraitul Ain Ali Haider**

**20053570019(3957)**

**CERTIFICATE**

This is to certify that the project entitled, “SARAL” has been done by: **Hardik Bhanot** and **Kuraitul Ain Ali Haider** of Bachelor of Science in Computer Science during semester IV from P.G.D.A.V.(M) College ,University of Delhi under the supervision of **Dr. APARNA DATT**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Aparna Datt

**DECLARATION**

We hereby declare that this Project Report titled “SARAL” submitted to the Department of Computer Science, P.G.D.A.V.(M) College, University of Delhi is a record of original work done by the team under the guidance of **Dr. Aparna Datt**.

The information and data given in the report is authentic to the best of the team’s knowledge.

This Project Report is not submitted to any other university or institution for the award of any degree, diploma or fellowship or published any time before.

**ACKNOWLEDGEMENT**

We would like to take this opportunity to express our profound gratitude and deep regards to our teacher Dr. Aparna Datt for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.

Our primary thanks to her, who poured over every inch of our project with painstaking attention and helped us throughout the working of the project.

It’s our privilege to acknowledge our deepest sense of gratitude to her for her inspiration which has helped us immensely. We are extremely grateful to her for unstilted support and encouragement in the preparation of this project.

**INDEX**

**1.Problem Statement 1**

(I) Problem definition

**2. Process Model 1-2**

(I) RAD model 1

(II) Advantages of RAD model 1

(III) Reasons for using RAD 2

**3.Requirements Analysis 2-4**

(I) SRS 3

(II) User Classes/functions 3

(III) Functional Requirements 4

**4.Use Case Diagram**

**5.Data Flow Diagram 6-9**

(I) Level 0 DFD 6

(II) Level 1 DFD 7

(III) Admin level 2 DFD 8

(IV) Customer level 2 DFD 8

(V) Tailor level 2 DFD 9

(VI) Delivery Assistant 9

**6.Entity Relationship Diagram 10**

**7.RDBMS Tables 11**

**8.Relational Database Executed in SQL 12-15**

**9.Timeline 16**

**10. Graphical User Interface 17-19**

(I) Hero Section 17

(II) Signup and Login 18

(III) Delivery 19

**11.Pseudocode 20**

(I) Delivery Module 20

(II) Order Module 20

**12. Flow Chart 21**

(I) Order Module 21

**13. Control Flow Graph(CFG) 22**

(I) Delivery Module 22

(II) Order Module 23

**14. Cyclomatic Complexity 24**

(I) Delivery Module 24

(II) Order Module 24

**15. Testing 25**

(I) Black Box Testing 25

(II) White Box Testing 25

**16.Test Cases 26**

(I) Login Module 26

**17. Bibliography 27**

**SARAL- Tailoring Made Easy**

**Problem Statement**

Currently, in the world of buying fashion online, the problem arises that the user can’t get accurate sized clothes. Also, tailors cannot go online due to the problem of having to take measurements physically or from old clothes.

To resolve this problem, we are building a platform from which people can determine and keep track of their body measurements and can use this for buying clothes online or can send it to their tailor. This also provides a better user interface as this technology is combined with A.R., users can try out the clothes virtually.

**SOFTWARE PROCESS MODEL**

Software Processes is a coherent set of activities for specifying, designing, implementing and testing software systems. A software process model is an abstract representation of a process that presents a description of a process from some particular perspective.

We have chosen the Rapid Application Development(**RAD**) Model.

**Rapid Application Development Model**

Rapid Application Development (RAD) model is a software development process based on prototyping without any specific planning. In the RAD model, there is less attention paid to the planning and more priority is given to the development tasks. It targets developing software in a short span of time.

**Advantages of Rapid Application Development Model**

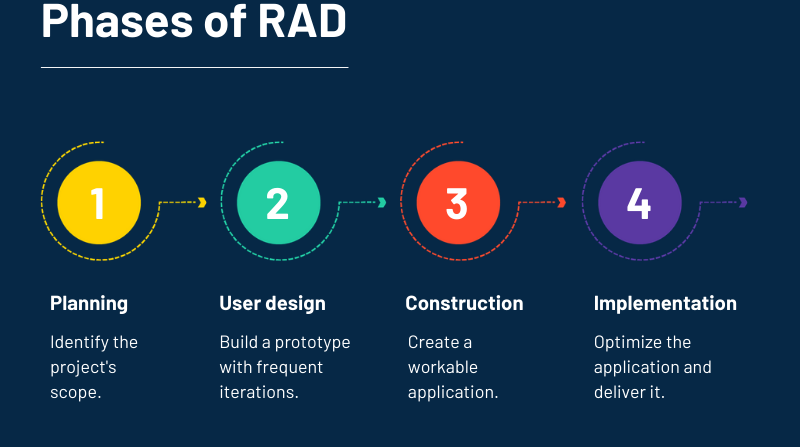
1. Increased customer satisfaction due to high-level collaboration and coordination between stakeholders (developers, clients, and end users).

2. Better risk management as stakeholders can discuss and address code vulnerabilities while keeping development processes going.

3. Quick iterations that reduce development time and speed up delivery.

4. Encouragement of code reuse, which means less manual coding, less room for errors, and shorter testing times.

2



**Reasons for using Rapid Application Development Model**

1. We need customer feedback constantly in our(This) Software as for improvement of the prototypes the customer feedback is important which increases the customer collaboration and coordination.
2. As tailors are also stakeholders in our(this) software they can provide input regarding the accuracy of the measurement that reduces the risk.
3. We(developers) need iterations for improving the accuracy of measurement in our software.
4. Since we(developers) are updating the code in iterations the code gets reused.

**REQUIREMENT ANALYSIS**

Requirement analysis helps organisations to determine the actual needs of stakeholders. At the same time, it enables the development team to communicate with stakeholders in a language they understand (like charts, models, flow-charts,) instead of pages of text.

Once the requirements are gathered, we document the requirements in a Software

Requirements Specification (SRS) document, use cases or as User Stories, which are shared with the stakeholders for approval. This document is easy to understand for both normal users and developers. Any changes in the requirements are also documented and go through a change control procedure and finalised on approval.

3

**Software Requirements Specification(SRS)**

The purpose of the software is to bridge the gap between tailors and users buying fashion online by providing a platform on which users can measure their body accurately online using which they can buy custom clothes from the tailor of their choice.

The key functionalities the project will include are:-

* The stakeholders of this project are Administrator, Tailors, Customer and delivery assistant(DA).
* Each user will have a login and signup module to use the functionality of the software.
* It should have a measurement module where customers can measure their body.
* Registration and maintenance of records of each category of user and transaction will be monitored and maintained by the administrator of the software.
* It should have an e-commerce module on which the tailor can list their designs,fabrics and the customer can place an order.
* It should have a chatting module for providing instructions between user and tailor.
* It should have a delivery module which is used by the delivery assistant for delivering the product from tailor to customer.
* It should have a tracking module for customers to track their orders.
* Each user will have their own dashboard, which will display the features of each user type.

**User classes and characteristics**

* **Administrators-** An administrator is an authorised officer of the company who has all the knowledge about the functionalities and processes of the company.

In this case a person/investor/team of this organisation is the administrator.

* **End Users-** In this project the end users are customer, tailors and Delivery Assistant.

Customers use this software for measuring their body and placing the order to their preferred tailor, while the tailor presents its designs and fabric to the customer using this platform. Delivery Assistant uses this to take finished product from the tailor and deliver it to the customer. Customers also access the tracking module.

* **Developers-** Developers are the members of the project team that are in charge of all of the System's graphical user interfaces. They, together with the administrator, will be responsible for the system's upkeep.

**Functional Requirements**

* **Login Module**

This module is accessed by the customer/tailor/administrator/Delivery Assistant for accessing their separate dashboards.

* **Measurement Module**

Here customers can click their image and upload it to the server where we process the image and will provide the measurements.

4

* **E-commerce Module**

Here, customers can choose to order from a wide range of tailors listed along with their designs and fabrics.

* **Chatting Module**

This module is accessed by the customer and tailor to get/provide specific instructions from the customer/tailor.

* **Delivery Module**

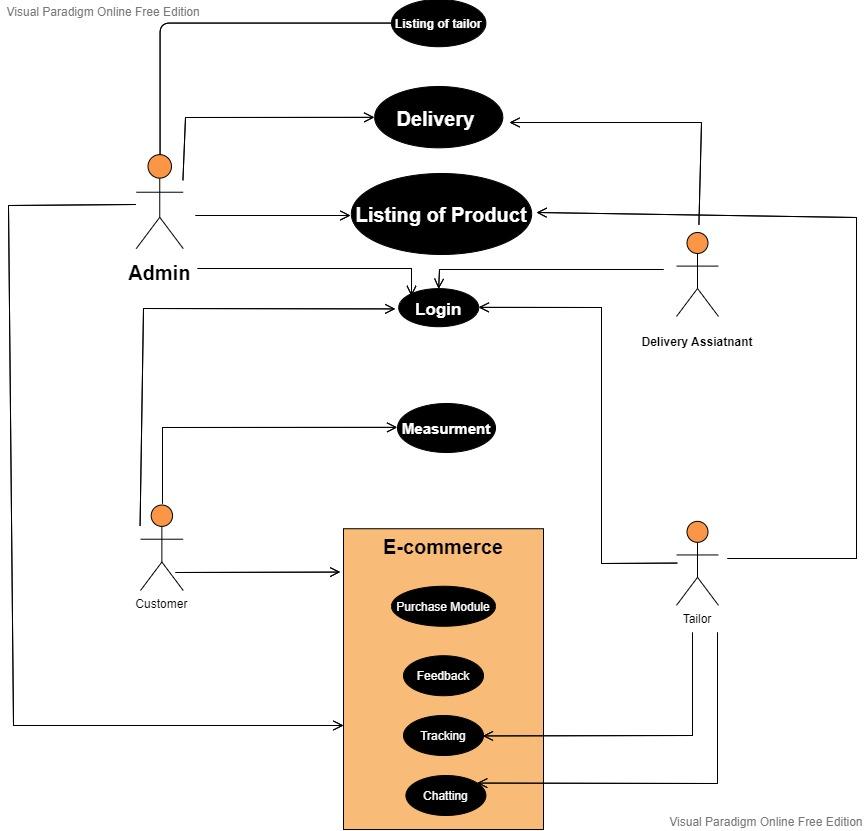
This module will be used for delivering the product from the tailor to the customer by Delivery Assistant.

* **Tracking Module-**

This module is used to track the product by the customer/tailor/administrator which is updated by Delivery Assistant.

5

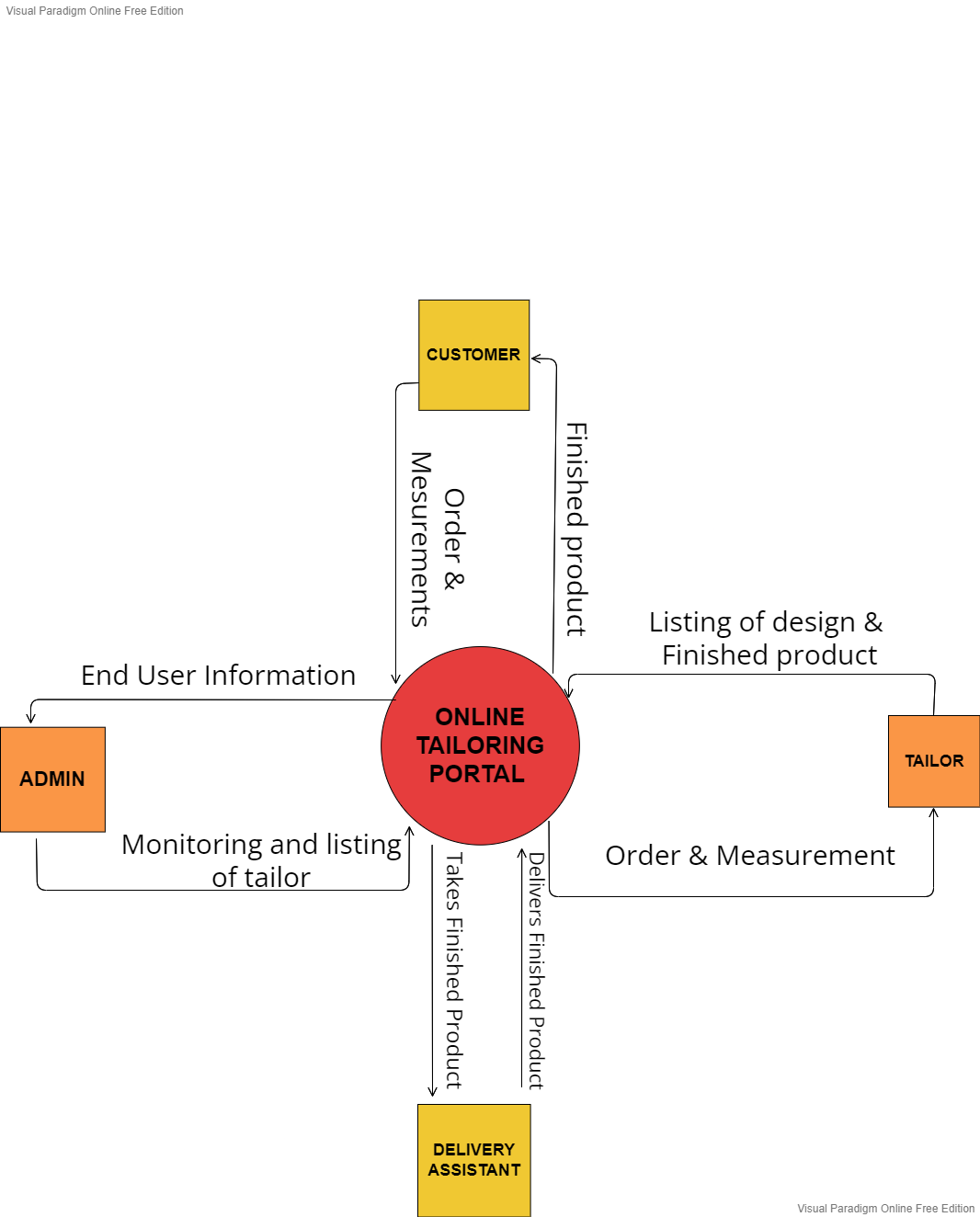
**Use Case Diagram**



6

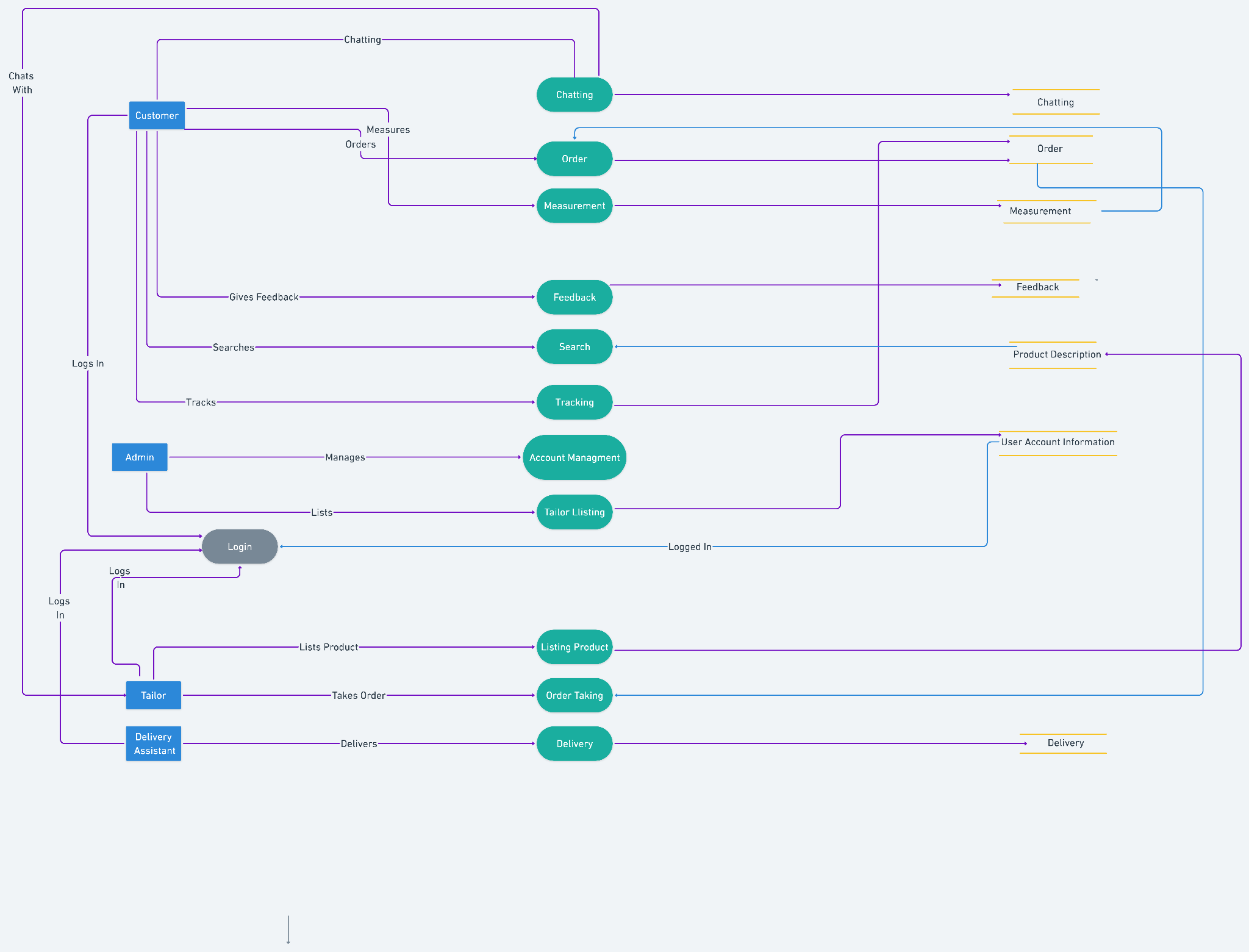
**Data Flow Diagrams(DFD)**

**Level 0 DFD**



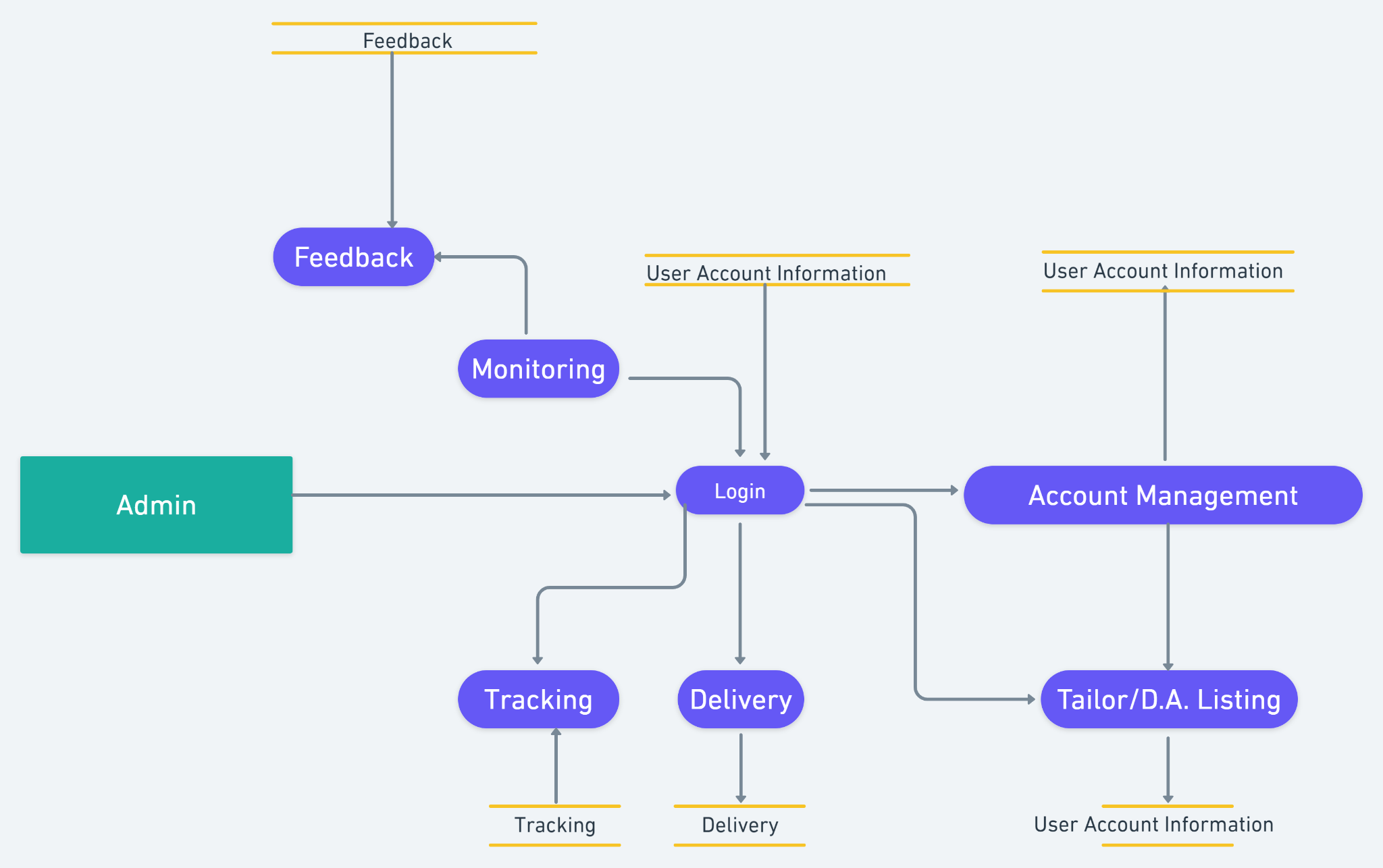
7

**Level 1 DFD**

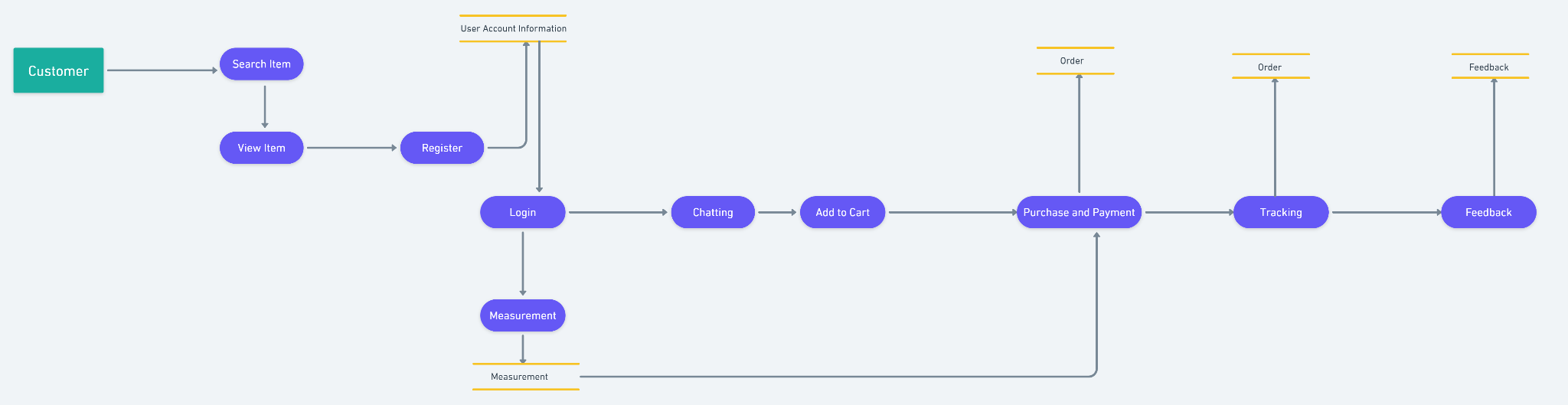


8

**Admin Level 2 DFD**

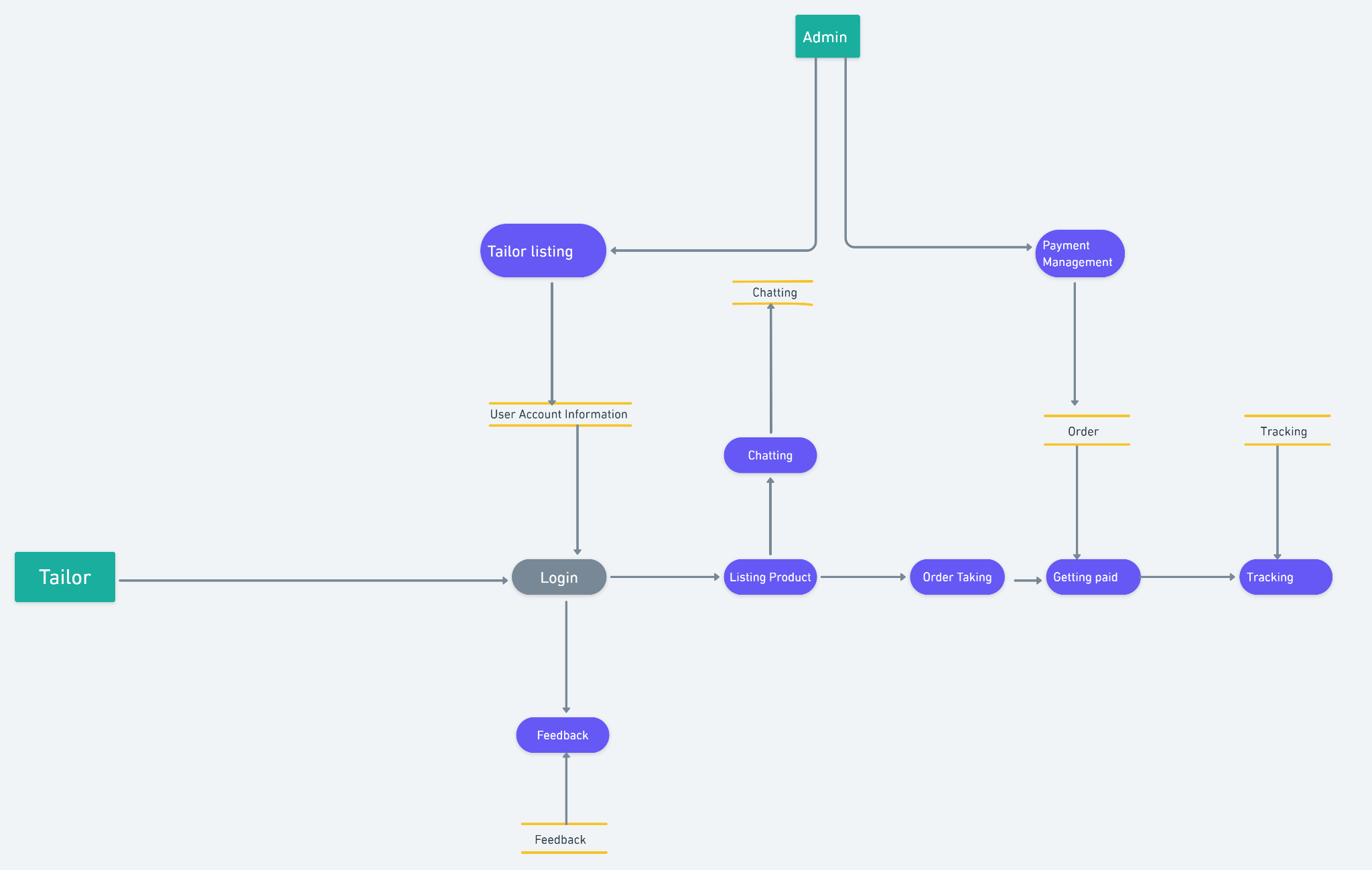


**Customer Level 2**

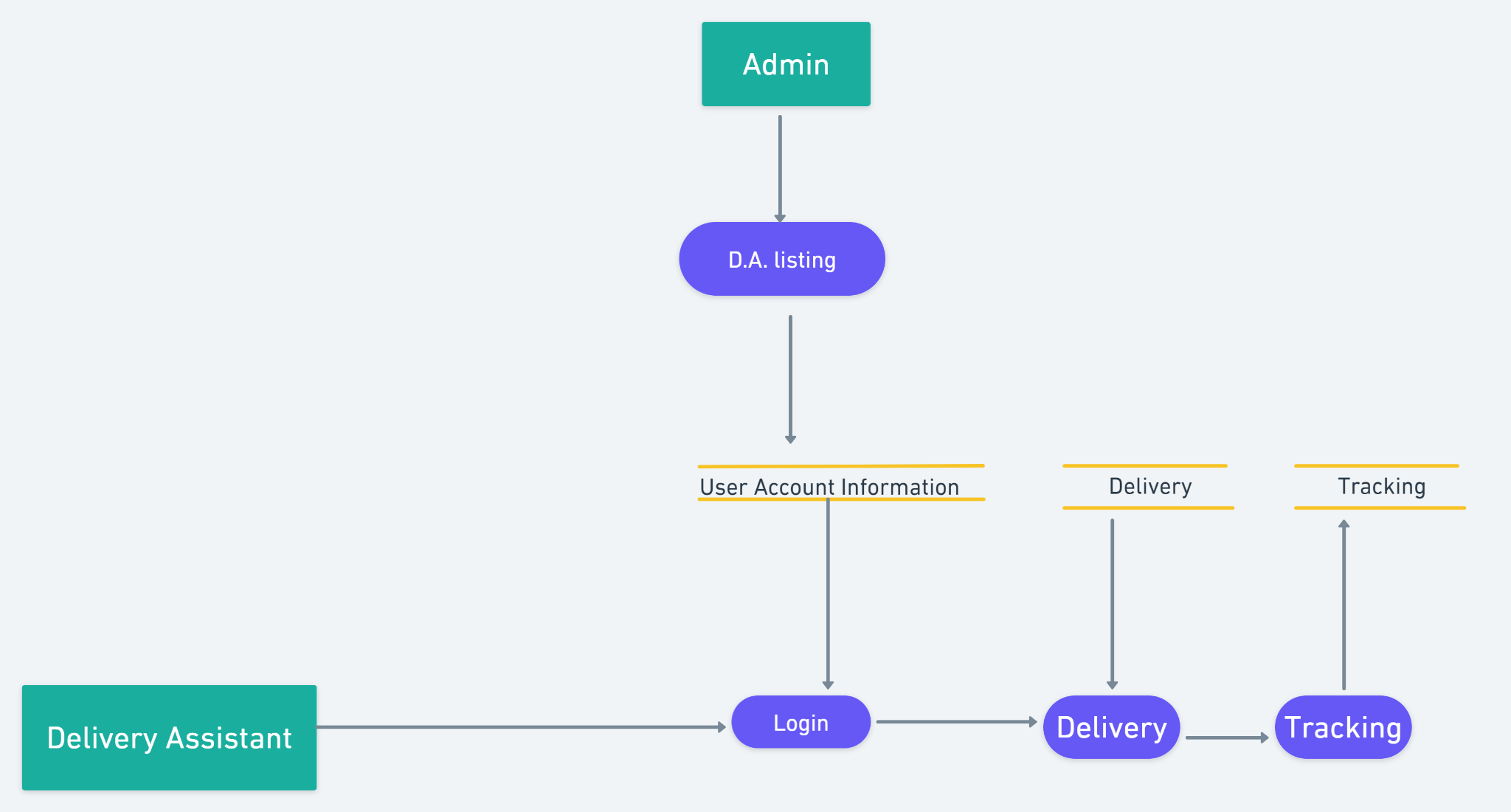


9

**Tailor Level 2 DFD**

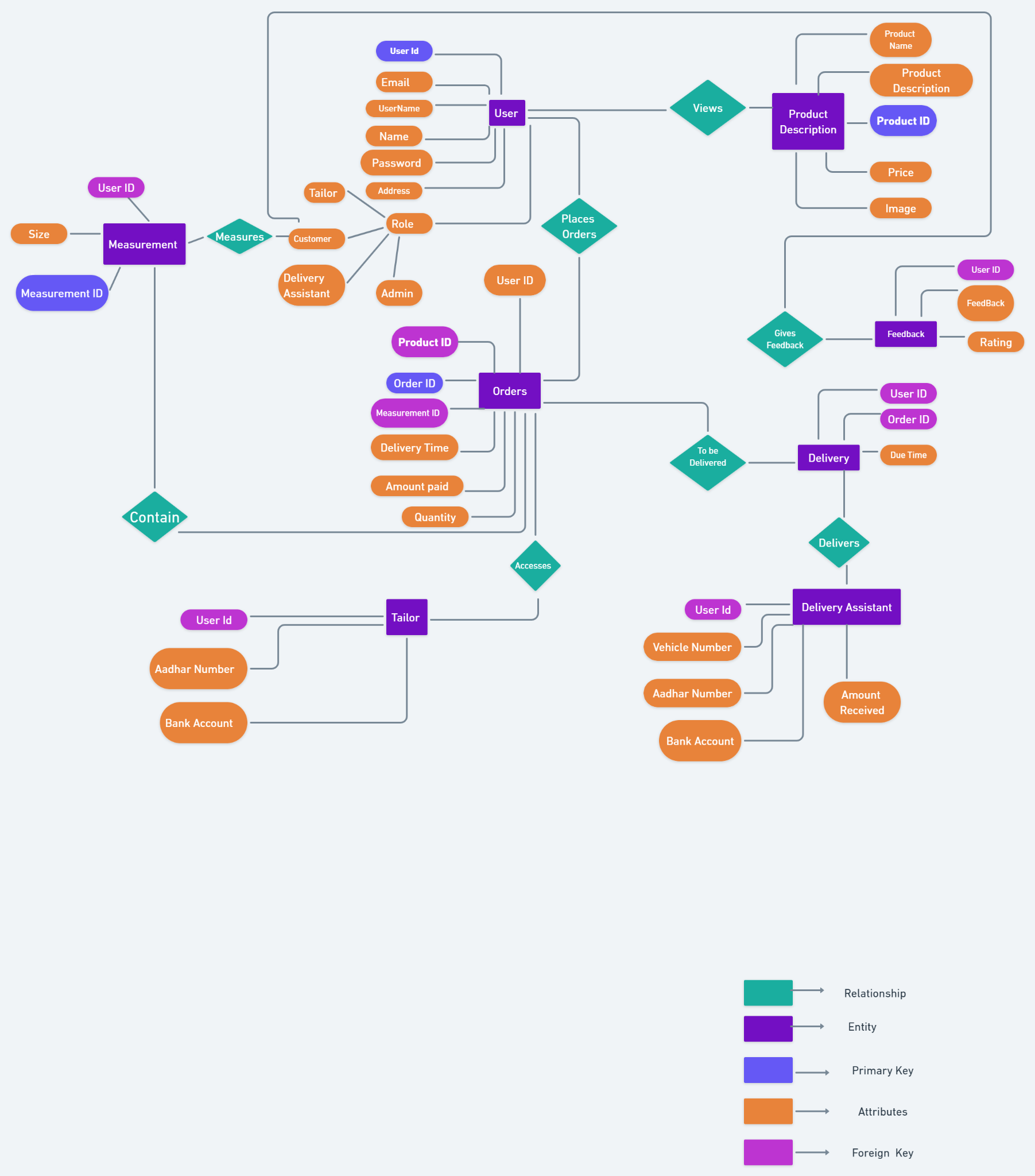


**Delivery Assistant Level 2 DFD**



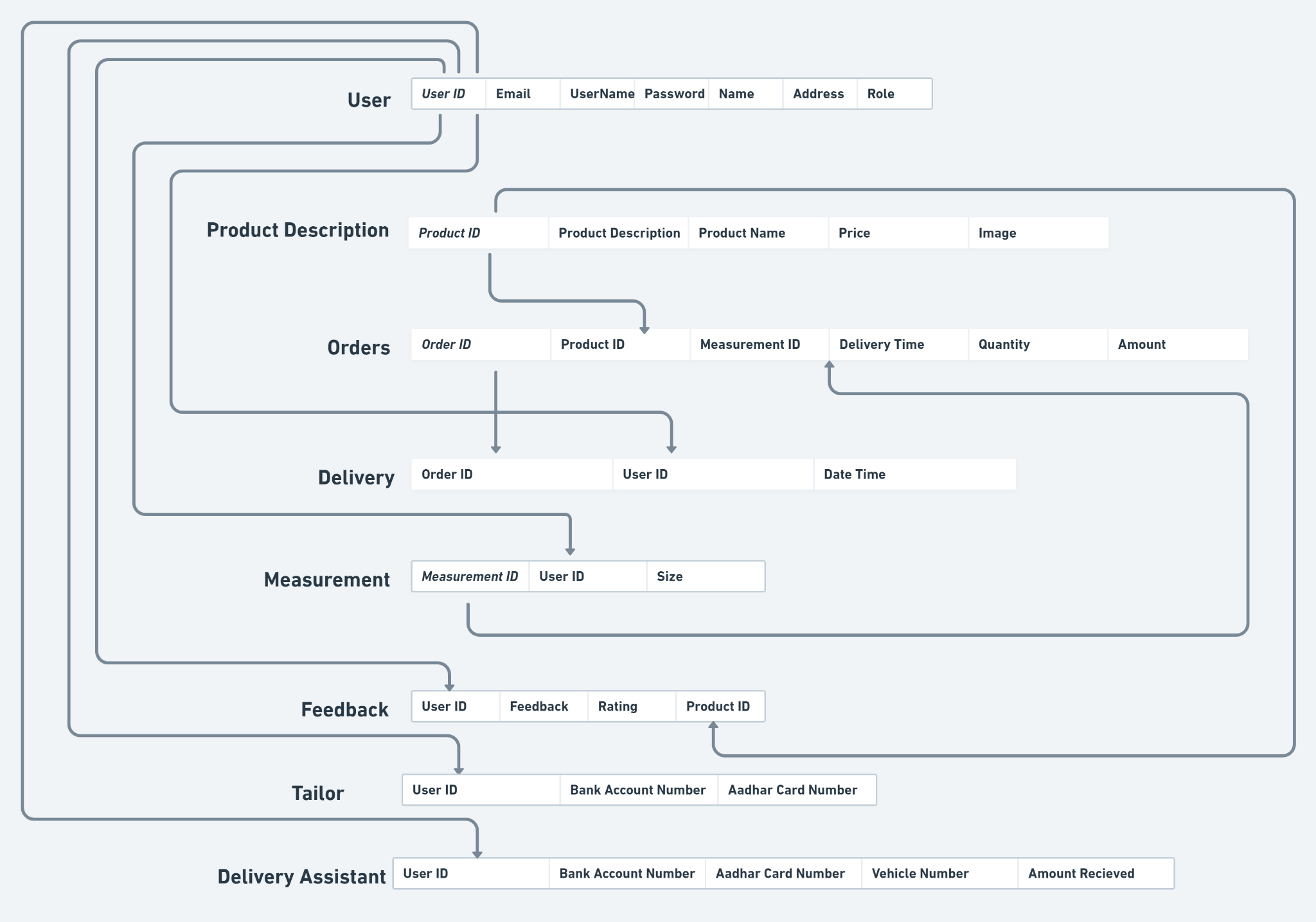
10

**Entity Relationship Diagram**



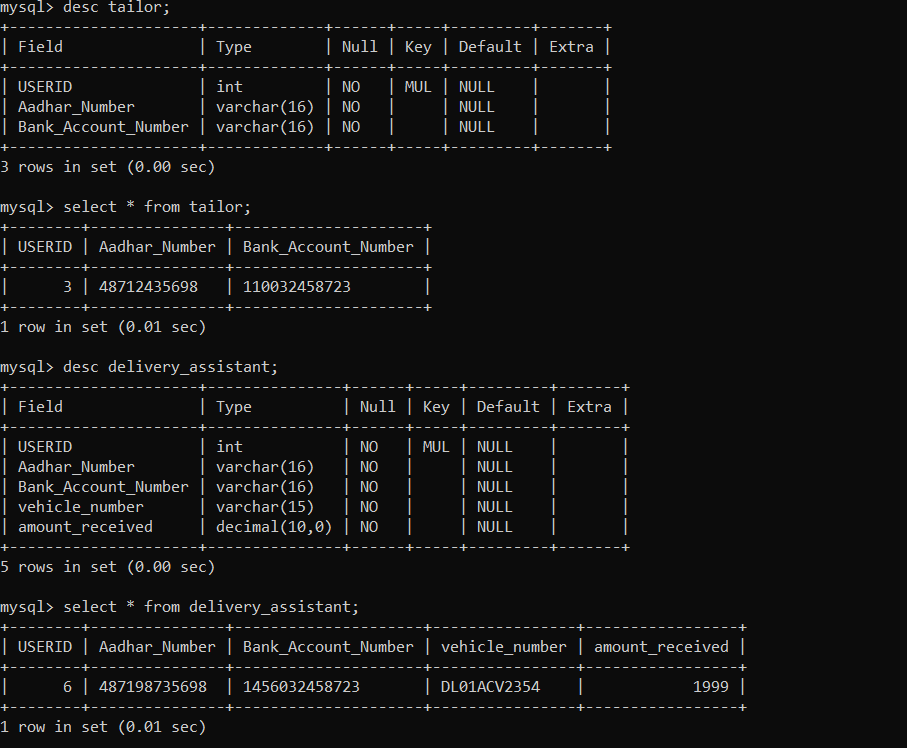
11

**Relational Database Management System(RDBMS)**



12

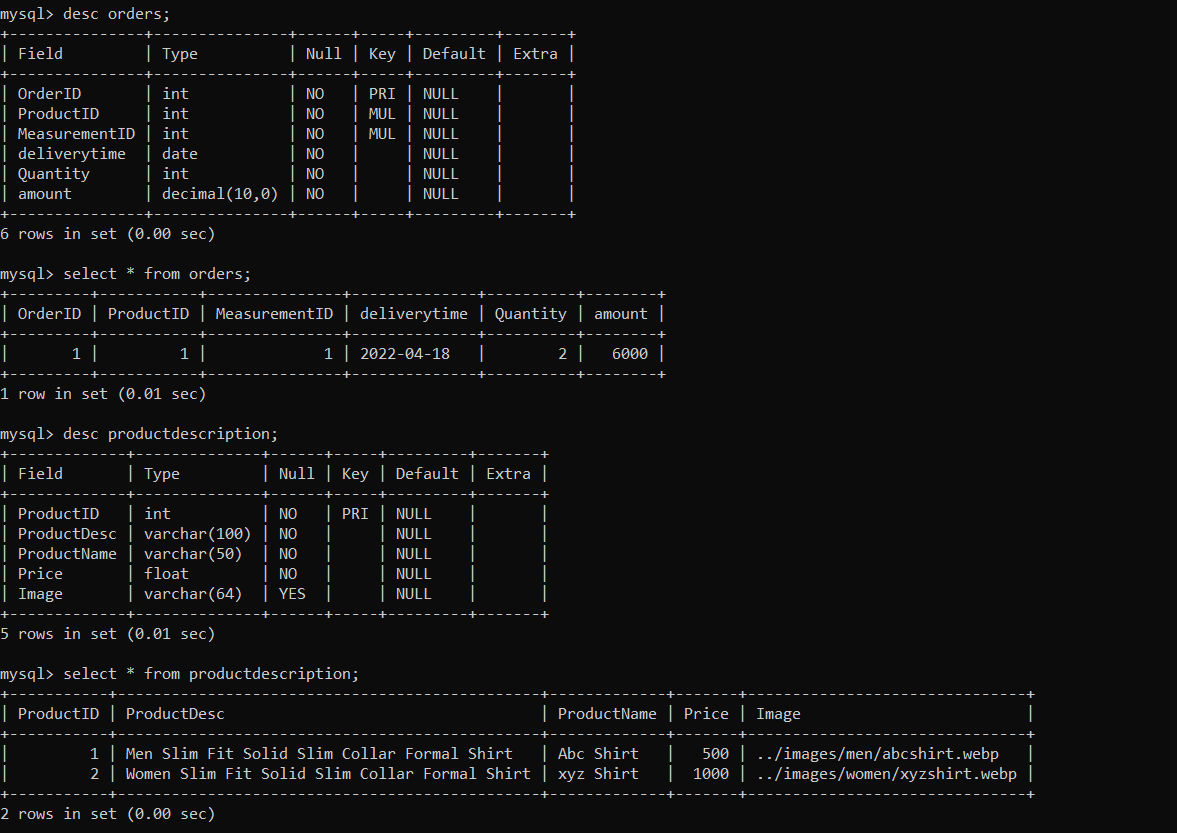
**Relational Database Executed in SQL**



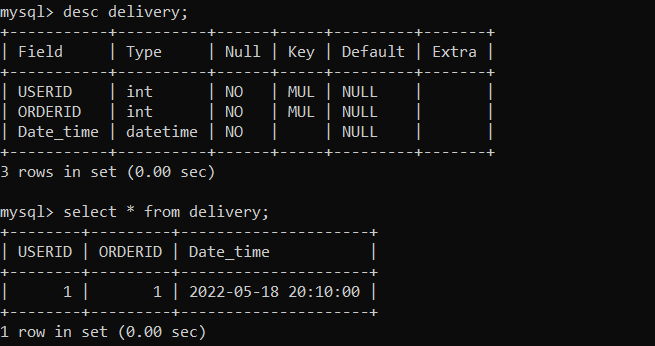
13



14

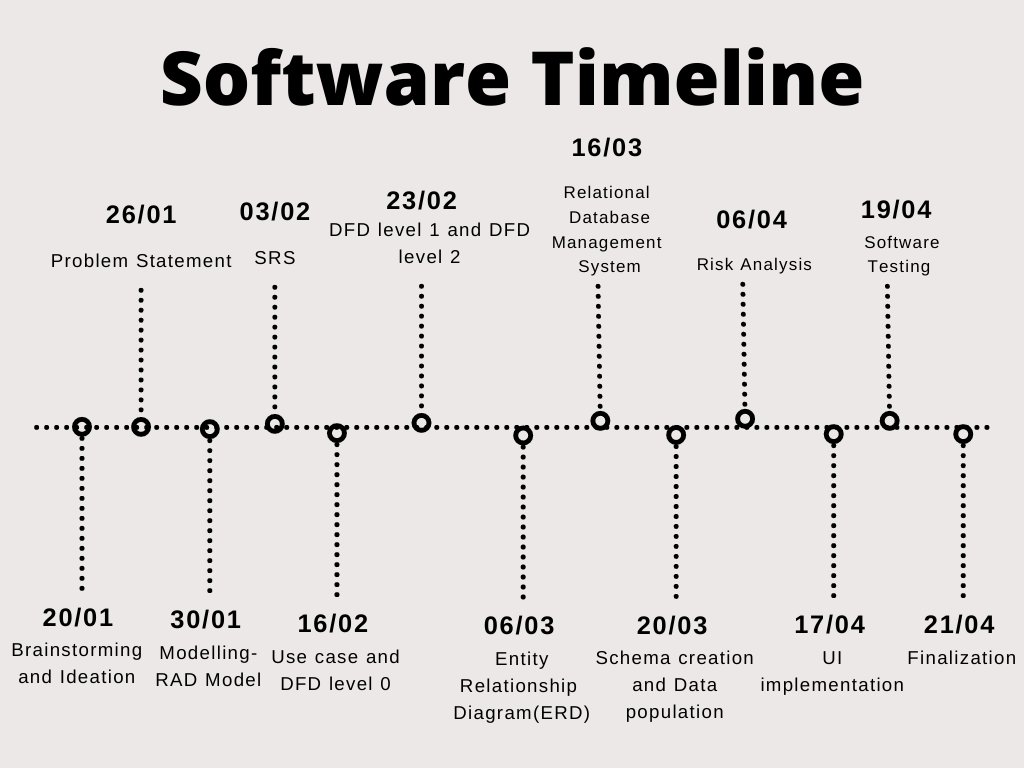


15



16

**Timeline**



17

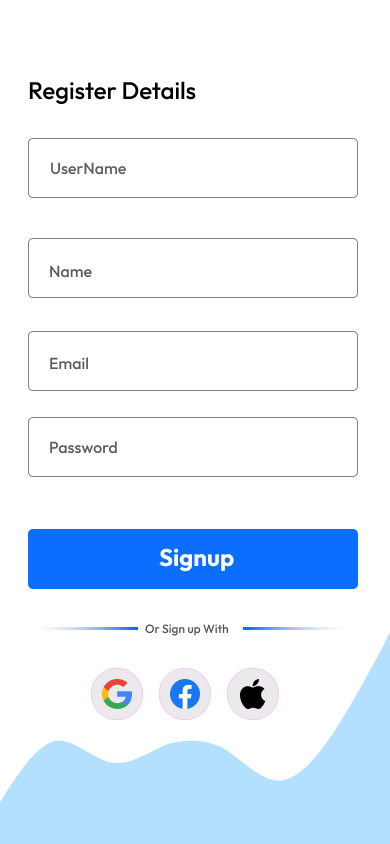
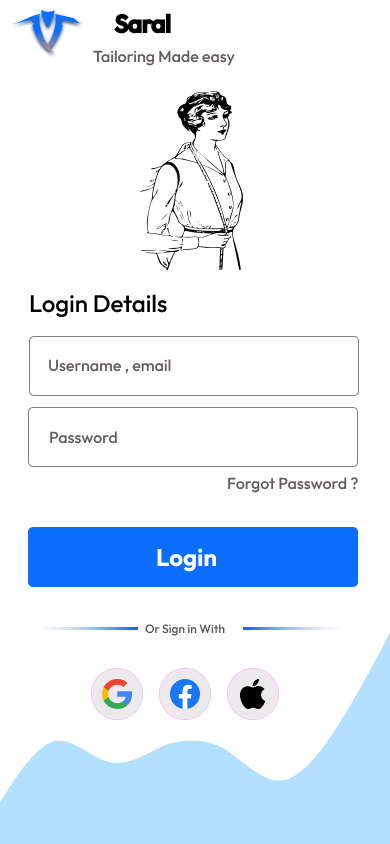
**Graphical User Interface (GUI)**

**Hero Section**



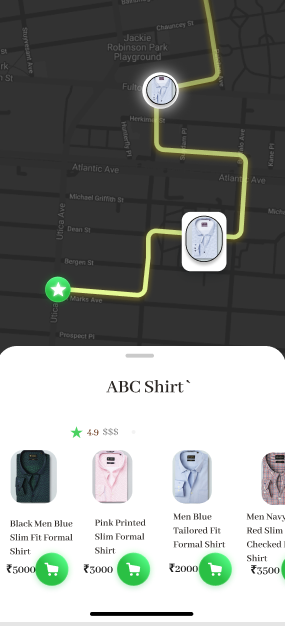
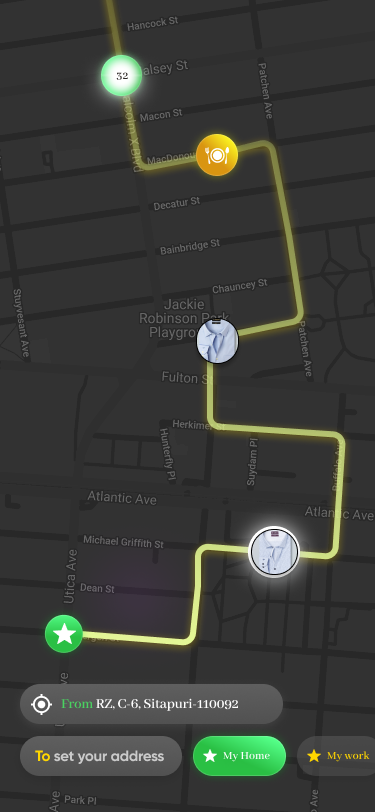
18

**Signup and Login**

19

**Delivery**

20

**Pseudocode**

Pseudocode is a [plain language](https://en.wikipedia.org/wiki/Plain_language) description of the steps in an [algorithm](https://en.wikipedia.org/wiki/Algorithm) or another system. Pseudocode often uses structural conventions of a normal [programming language](https://en.wikipedia.org/wiki/Programming_language), but is intended for [human](https://en.wikipedia.org/wiki/Human) reading rather than machine reading. It typically omits details that are essential for machine understanding of the algorithm, such as [variable declarations](https://en.wikipedia.org/wiki/Variable_declaration) and language-specific code.

**Delivery Module**

1.order=database.get(orders.ID=deliveryorder.ID)

2.user=database.get(delivery.userID=user.userID)

3.available\_delivery\_boy=database.get(delivery.assistant.status=available

4.product.setDelivery(address:user.address,amount:order.amount,available\_delivery\_boy)

**Order Module**

1. user=session.get()
2. product= user.select()
3. payment\_method=user.select()
4. if(payment\_method==”online”){
5. payment\_details=user.input()

}else{

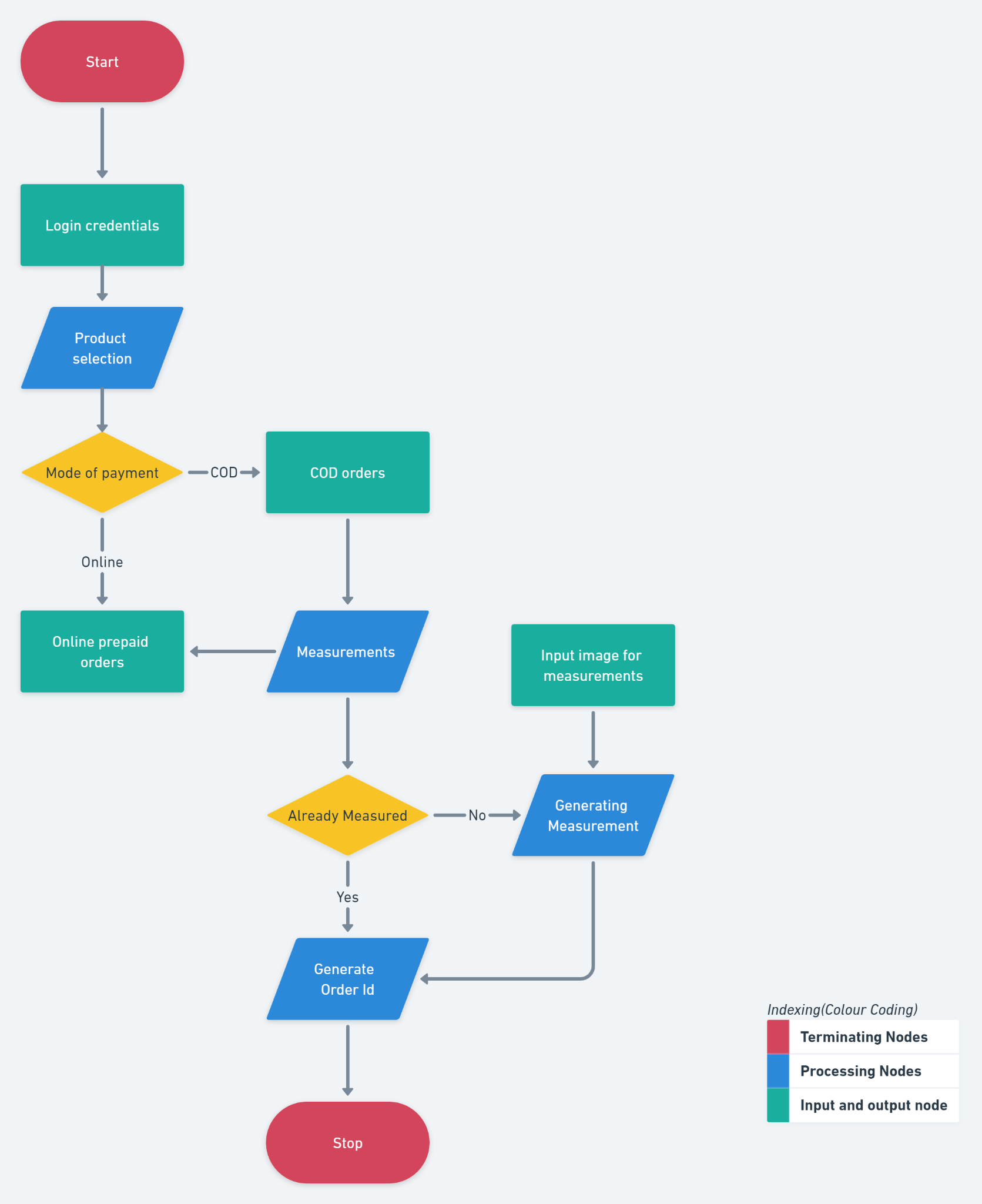
1. payment\_method\_set(“Cash on Delivery”)}
2. measurement=database.get(measurement.usedID==user.userid)
3. if(measurement==NULL){
4. measurement==user.input()}
5. else{get.measurement.size)
6. generate orderID

21

**Flow Chart**

A [flowchart](https://www.edrawsoft.com/flowchart-definition.html) is the graphical or pictorial representation of an algorithm with the help of different symbols, shapes, and arrows to demonstrate a process or a program. With algorithms, we can easily understand a program. The main purpose of using a flowchart is to analyse different methods.

**Order Module**

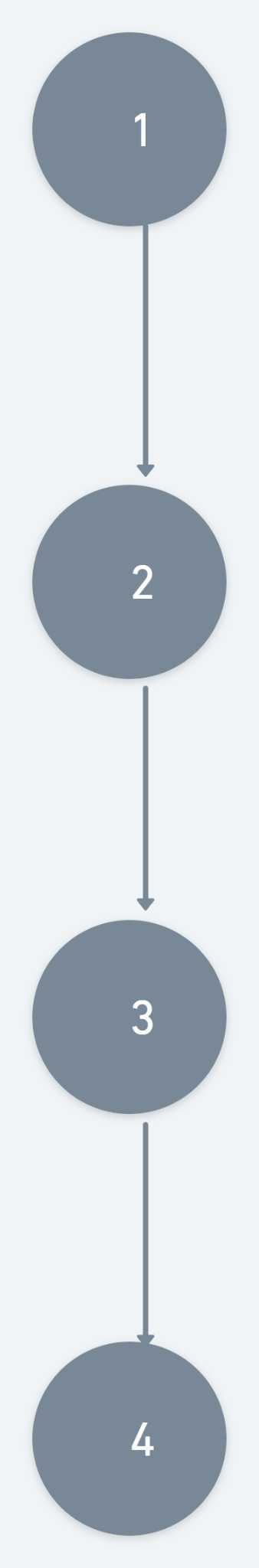


22

**Control Flow Graph**

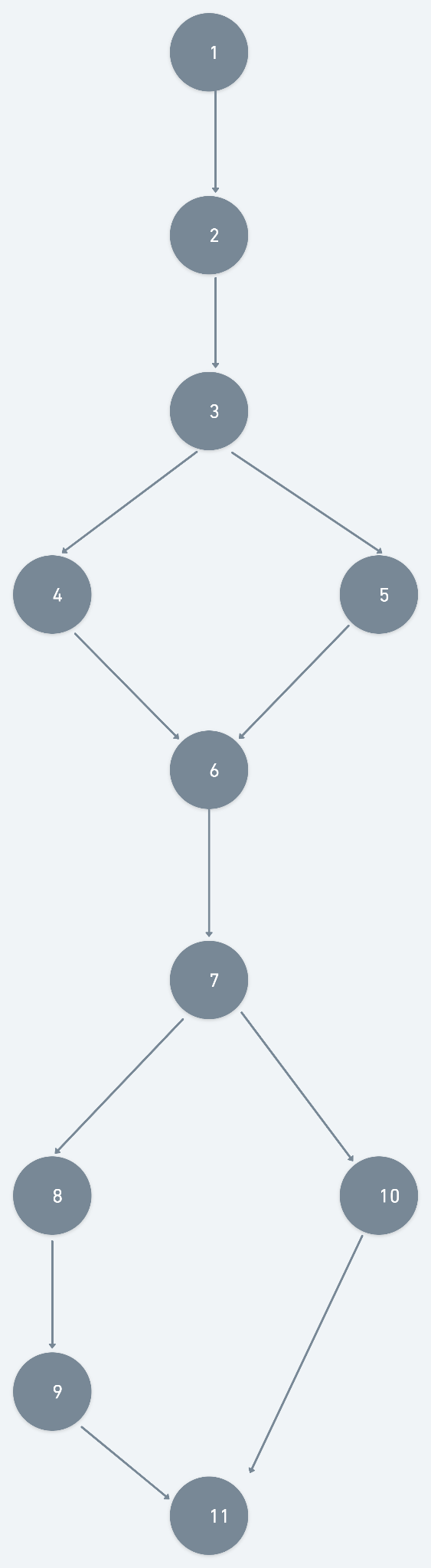
A Control Flow Graph (CFG) is the graphical representation of control flow or computation during the execution of programs or applications. Control flow graphs are mostly used in static analysis as well as compiler applications, as they can accurately represent the flow inside of a program unit.

**Delivery Module**

****

23

**Order Module**



24

**Cyclomatic Complexity**

Cyclomatic complexity is a [software metric](https://en.wikipedia.org/wiki/Software_metric) used to indicate the [complexity of a program](https://en.wikipedia.org/wiki/Programming_complexity). It is a quantitative measure of the number of linearly independent paths through a program's [source code](https://en.wikipedia.org/wiki/Source_code).It is computed using the [control-flow graph](https://en.wikipedia.org/wiki/Control-flow_graph) of the program: the nodes of the [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)) correspond to indivisible groups of commands of a program, and a [directed](https://en.wikipedia.org/wiki/Directed_graph) edge connects two nodes if the second command might be executed immediately after the first command. Cyclomatic complexity may also be applied to individual [functions](https://en.wikipedia.org/wiki/Function_(computer_science)), [modules](https://en.wikipedia.org/wiki/Modular_programming), [methods](https://en.wikipedia.org/wiki/Method_(computer_science)) or [classes](https://en.wikipedia.org/wiki/Class_(computer_science)) within a program.

Cyclomatic Complexity = Number of Edges - Number of Nodes +2

**Delivery**

Number of Edges = 3

Number of Nodes = 4

Cyclomatic Complexity(Delivery) = 3-4+2 = 1

Number of closed regions = 0

Cyclomatic Complexity(Delivery) = 1+0(closed region) =1

**Orders**

Number of Edges = 12

Number of Nodes = 11

Cyclomatic Complexity(Orders) = 12-11+2 = 3

Number of closed regions = 2

Cyclomatic Complexity(SignUp) = 1+2(closed region) =3

25

**TESTING**

* The process of executing a programme with the goal of identifying an error is known as testing. A good test case has a high chance of uncovering a mistake that has yet to be identified.
* Software testing is an important part of software quality assurance since it is the final check of the specification, design, and code.
* Product testing is used to check and validate various work products, such as software. to guarantee that the modules, integrated unit, and finished product all satisfy the specifications.

**Black Box Testing:**

* Functional testing is another name for black box testing.
* Black Box Testing is a test case design process that focuses on the software's functional needs and allows a software engineer to generate a set of input circumstances that completely exercise all of the program's functional requirements.
* Examine the objects from an outside perspective.
* Specifications are used to test data, such as what type of input should be sent to the unit or module.
* We can test the functioning based on the output and input, rather than looking at the inside coding.
* It attempts to find errors in the following categories

1. Incorrect or missing functions
2. Interface errors
3. Errors in data structure or External database access
4. Behaviour or performance error
5. Initialisation and termination errors

**White Box Testing:**

* Glass box testing is another name for it.
* White Box testing is a test case design approach that derives test cases from the procedural design's control structure.
* Using White Box Testing method, the software engineer can derive test cases that
* Guarantee that all independent paths within a module have been exercised at least once.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise internal data structures ensure their validity.
* Test the artefacts from the internal point of view.
* It cannot detect absence of features.
* For security purposes the Email of the user is required in case he/she forgets his/her password and wants to retrieve that.

26

**Test Cases**

**LogIn Module:**

| **S.No.** | **Test Case** | **Expected Result** |
| --- | --- | --- |
| 1 | Valid Email/Username and Valid Password | LogIn Successful |
| 2 | Valid Email/Username and Invalid Password | LogIn Unsuccessful |
| 3 | Invalid Email/Username and Valid Password | LogIn Unsuccessful |
| 4 | Non Existing Email/Username and Invalid Password | LogIn Unsuccessful |

27

**BIBLIOGRAPHY**

* Software Engineering – A Practitioner’s Approach o By Roger S. Pressman
* Software Engineering- K.K. Aggarwaland Yogesh Singh
* <https://www.javatpoint.com/software-engineering-sdlc-models>
* <https://www.javatpoint.com/software-engineering-rapid-application-development-model>
* <https://www.geeksforgeeks.org/types-software-testing/>