Lab Manual

Experiment 1: Write the problem statement, its purpose, scope, and literature review. The problem statement is intended for a broad audience and should be written in non-technical terms.

Project Title: Pharmacy Management System

Introduction

Project Summary

The software is designed for the use of record keeping. Pharmacy Management system includes the feature of keeping the record of drugs in supply, in sale, expired, the employee details, the sales and financial details and a simple interface to interact for the employee. Not all employees have the privilege to access it, certain employees in certain departments with certain designations are permissioned to access and use it. The system is an efficient way to hold the record of everything that is of value to a Pharmacy; for example the drugs data, the inventory status of each drug, the supplier information, the customer review etc.

Purpose

As the environmental conditions are getting worse every single year, it has become a necessity to have a medicine store at every corner of the world. Pharmacy has become a fundamental part of normal human life. Like everything else, pharmacies should be managed in an efficient way and have to have a certain system to operate the demand, feedback and sales so that the sole focus of the pharmacy and its employees will be the service of the customer as soon and as best as possible.
The purpose of this project is to design that system so that it can operate as a third vital role in a company beside employees and customers.

The objective of this project includes

1. Build software to help manage a pharmacy
2. Keep track of drugs
3. Take customer feedback and use it for sales strategy
4. Storing the employee information, the financial information and sales information
5. Keeping track of the purchase and sales of each drugs in store along with the demand
6. Make the software easy and attractive to use

Scope

1. Provide free access to users
2. Provide the user with a list of functionalities
3. An interface for each function to the user

Technology and Literature Review of past work

Technology:

Front End: PHP, HTML,CSS
Back End: PHP, MYSQL

Literature Review:

1. **Cerner Retail Pharmacy:** Cerner Retail Pharmacy runs on a proven, time-tested, and easy-to-use Windows-based platform. It efficiently improves workflow, increases profits, and lowers operating costs.

2. **McKesson Pharmacy Systems:** It is a cloud-based pharmacy management system that provides cloud support,online order of drugs and consultation

Experiment 2: Develop a requirements specification for the above problem (The requirements specification should include both functional and non-functional requirements) and also includes the constraints.

Solution:

System Requirements Study:

1. Introduction

1.1 Purpose

As the environmental conditions are getting worse every single year, it has become a necessity to have a medicine store at every corner of the world. Pharmacy has become a fundamental part of normal human life. Like everything else, pharmacies should be managed in an efficient way and have to have a certain system to operate the demand, feedback and sales so that the sole focus of the pharmacy and its employees will be the service of the customer as soon and as best as possible.
The purpose of this project is to design that system so that it can operate as a third vital role in a company beside employees and customers.

1.2 Document Conventions

The document uses the following conventions:

DB	Database
ER	Entity Relationship
DFD	Data Flow Diagram

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the pharmacy management system and is intended to be used by the pharmacy owners and employees. The system can be used by the employees to keep track of drug, sales and customer feedback.

1.4 Product Scope

1. Provide access to users who are eligible
2. Provide the user with a list of functionalities
3. An interface for the user that is easy to use.

1.5 References

https://en.wikipedia.org/wiki/
http://vlabs.iitkgp.ernet.in/se/
https://www.lucidchart.com
https://www.geeksforgeeks.org
https://www.perforce.com

2 Overall Description

2.1 Product Perspective

The product perspective is to make software that can be easily used by pharmacy owners and employees to keep track of drug purchase, sale and customer feedback along with other information such as financial information.

2.2 Product Functions

The functionality of the product includes the following:

1. Users are able to login into the system
2. Users can update the details about the medicines
3. They can keep track of the number of medicines supplied by the suppliers.
4. They can keep track of the sales going on.
5. They can keep track of the number of medicines remaining in the stock.
6. They can keep track of the drug purchase from different companies
7. They can keep track of the employees and finance
8. They can store and analyze the customer response.

2.3 Operating Environment

Operating Environment for the application is enlisted below:

1. client/server system
2. OS: Windows, Linux, Mac

2.3 Design and Implementation Constraints

The pharmacy management system is dedicated to the medicines retailers. Sometimes the demand made by those retailers in the software can be unclear which can create confusion and can cost them more. So the user should clear their requirements before system implementation.

2.4 User Documentation

The user will be able to know more about the application via a guide provided by the developers along with the application and can also refer to the application's online social media platforms.

3 External Interface Requirements

3.1 User Interfaces

1. Front-End: HTML, CSS, javascript, jquery
2. Back-End: ajax, Php, Mysql

3.2 Hardware Interfaces

1. Windows 10,11 running hardware

3.3 Software Interfaces

Software Used	**Description**
Php	Php is a widely-used programming language for back-end

	development.
javascript	It is a programming language used at the front end so as to make the application more user interactive and friendly.
Operating System	The chosen OS is windows as it is used by a wide range of people across the world. But the application works on all OS.

3.4 Communications Interfaces

We are using HTTPS to connect to the application and FTP to upload files to the FTP server. We are also using SMTP and POP3 for mail delivery purposes.

4 System Features

The different features of the system are included below:

4.1 Login:

4.1.1 Description and Priority

This functionality allows the user to log into the software. It has a higher priority.

4.1.2 Stimulus/Response Sequences:

1. Username / password
2. if correct / valid redirect to dashboard.

4.1.3 Functional Requirements:

1. **REQ-1:** Username/Password

   It helps login into the pharmacy management system dashboard, from where users can perform actions according to their privilege.

2. **REQ-2:** Client/Server System

   The term client/server refers primarily to architecture or logical division of responsibilities, the client is the appliAll the applications execute on the client side. cation (also known as the front-end), and the server is the DBMS (also known as the back-end)

   - All the data resides at the server-side (model)

- All the applications execute on the client side.

4.2 Purchase

4.2.1 Description and Priority

This functionality directly allows the pharmacy owner to purchase or order medicines from its contracted medicine manufacturer. It really helps during a time of less stock of medicine. It has a higher priority.

4.2.2 Stimulus/Response Sequences:

1. Purchase medicine
2. Acceptance or Rejection of order by the manufacturer.

4.2.3 Functional Requirements:

1. **REQ-1:** Logged In

   The user needs to be logged into the pharmacy owner account in order to make the purchase or order medicine.

2. **REQ-2:** Client/Server System

   The term client/server refers primarily to architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end)

   - All the data resides at the server-side (model)
   - All the applications execute on the client side.

4.3 Sell

4.3.1 Description and Priority

This functionality directly allows the pharmacy to sell their medicine in other words it is purchased or bought for the customers. Customers can use this feature to buy medicine. Internally the functionality is made available as a sell with respect to the pharmacy owner.

4.3.2 Stimulus/Response Sequences:

3. Purchase medicine
4. Acceptance or Rejection of order by the manufacturer.

4.3.3 Functional Requirements:

3. **REQ-1:** Logged In

   The user needs to be logged into the pharmacy user account in order to make the purchase or order medicine.

4. **REQ-2:** Client/Server System

   The term client/server refers primarily to architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end)

   1. All the data resides at the server-side (model)
   2. All the applications execute on the client side.

4.4 Sales

4.4.1 Description and Priority

This functionality directly allows the pharmacy owner or authorized employee to keep track of the sales of the medicine. The selling can be through online orders as well as offline retail sales. It really helps get accurate information on the profit made by the pharmacy.

4.4.2 Stimulus/Response Sequences:

1. Sell medicine
2. Keep records of sold medicine value in the database

4..3 Functional Requirements:

1. **REQ-1:** Logged In

   The user needs to be logged into the pharmacy owner account or authorized employee account in order to make the change in the sales table.

2. **REQ-2:** Client/Server System

The term client/server refers primarily to architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end)

1. All the data resides at the server-side (model)
2. All the applications execute on the client side.

4.5 Customer Feedback

4.5.1 Description and Priority

This functionality directly allows the customer to give their feedback to the pharmacy about its service and quality of medicine. It has a medium priority.

4.5.2 Stimulus/Response Sequences:

1. Purchase medicine
2. Give feedback about the medicine and pharmacy.

4.5.3 Functional Requirements:

5. **REQ-1:** Logged In

   The user needs to be logged into the pharmacy account in order to make the purchase or order medicine and give their reviews.

6. **REQ-2:** Client/Server System

   The term client/server refers primarily to architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end)

   1. All the data resides at the server-side (model)
   2. All the applications execute on the client side.

5 Other Nonfunctional Requirements

5.1 Safety Requirements

The following actions will be taken to reduce risks in this project; AWS cloud services to backup code and prevent the loss, use own initiative to keep on top of the project plan, regularly make backups of project reports and important documents, if an unplanned day off occurs to make up for work lost, a backup database for data saving incase of hacks, natural disaster, fire, etc,

5.2 Security Requirements

There is always a chance of hackers intercepting the query request for the user and modifying it to be something else. Also, there is a high chance of making a mistake in the code which might be a bug in the future and can be exploited by hackers.

5.3 Software Quality Attributes

AVAILABILITY: Every feature of the application must be available at all times whenever any user fires the application.

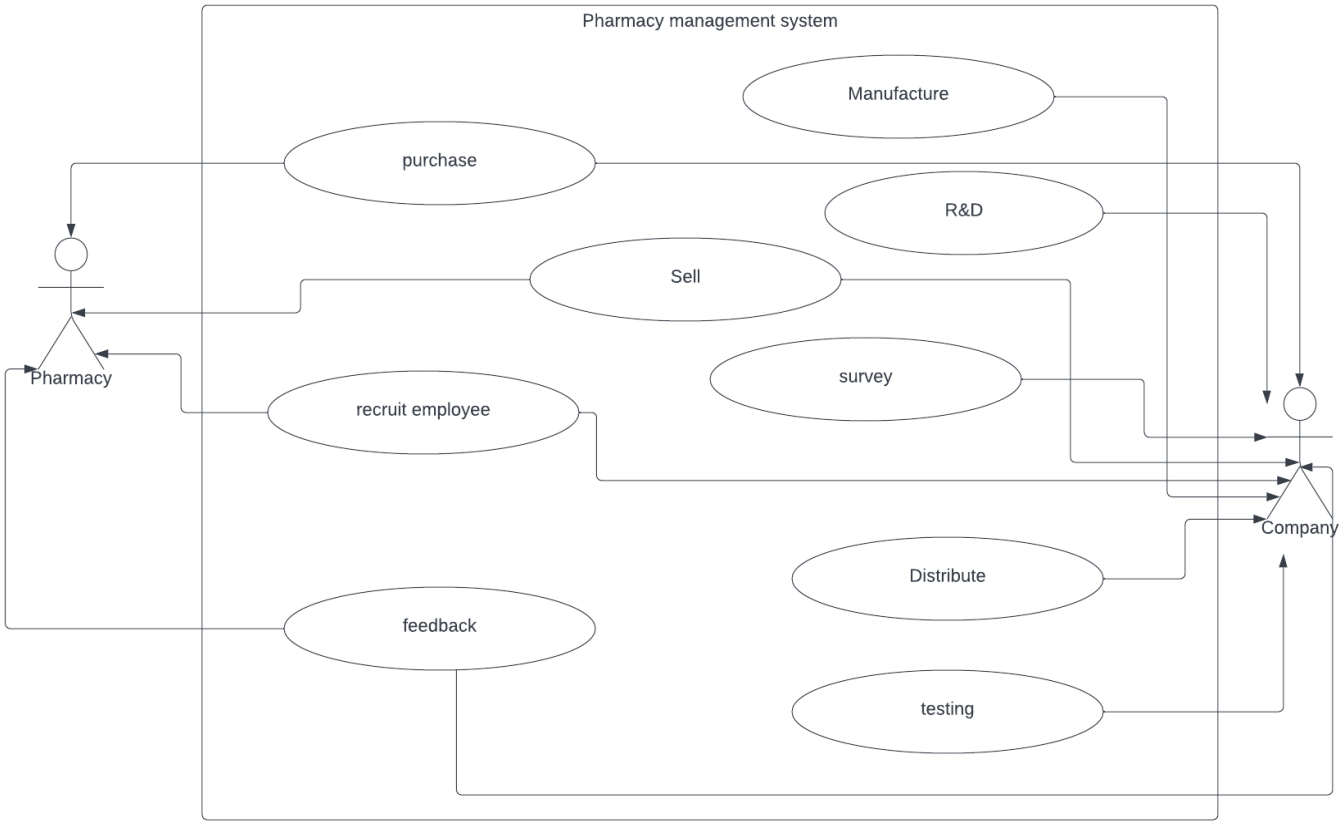
CORRECTNESS: The user should be able to get the correct information from the query.

MAINTAINABILITY: The developers should be able to maintain the quality of the application to do so updates can be implemented on the application.

USABILITY: The application should be able to satisfy the requirements of the users.

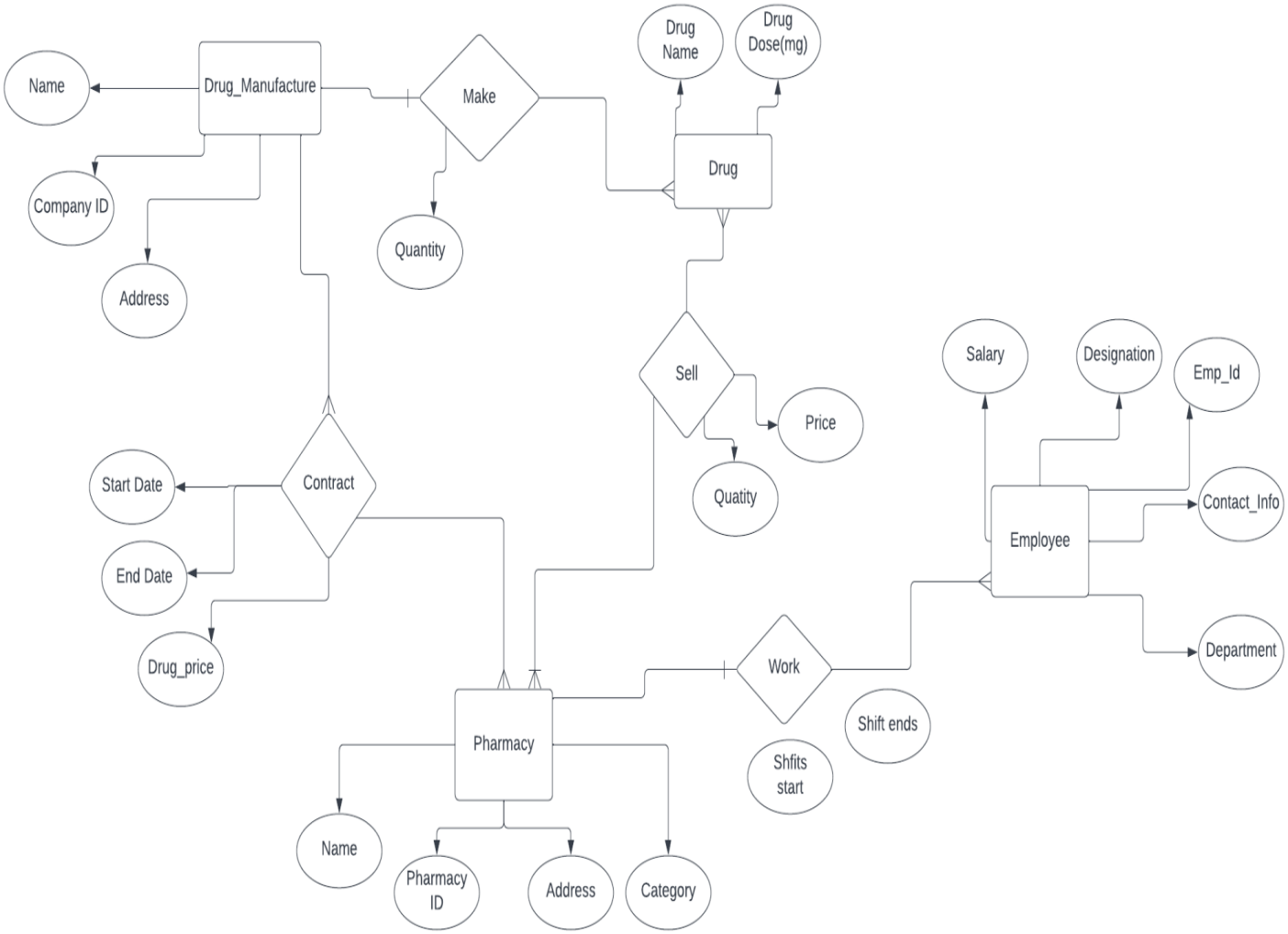
Experiment 3: To perform the user's view analysis for the suggested system

Solution:



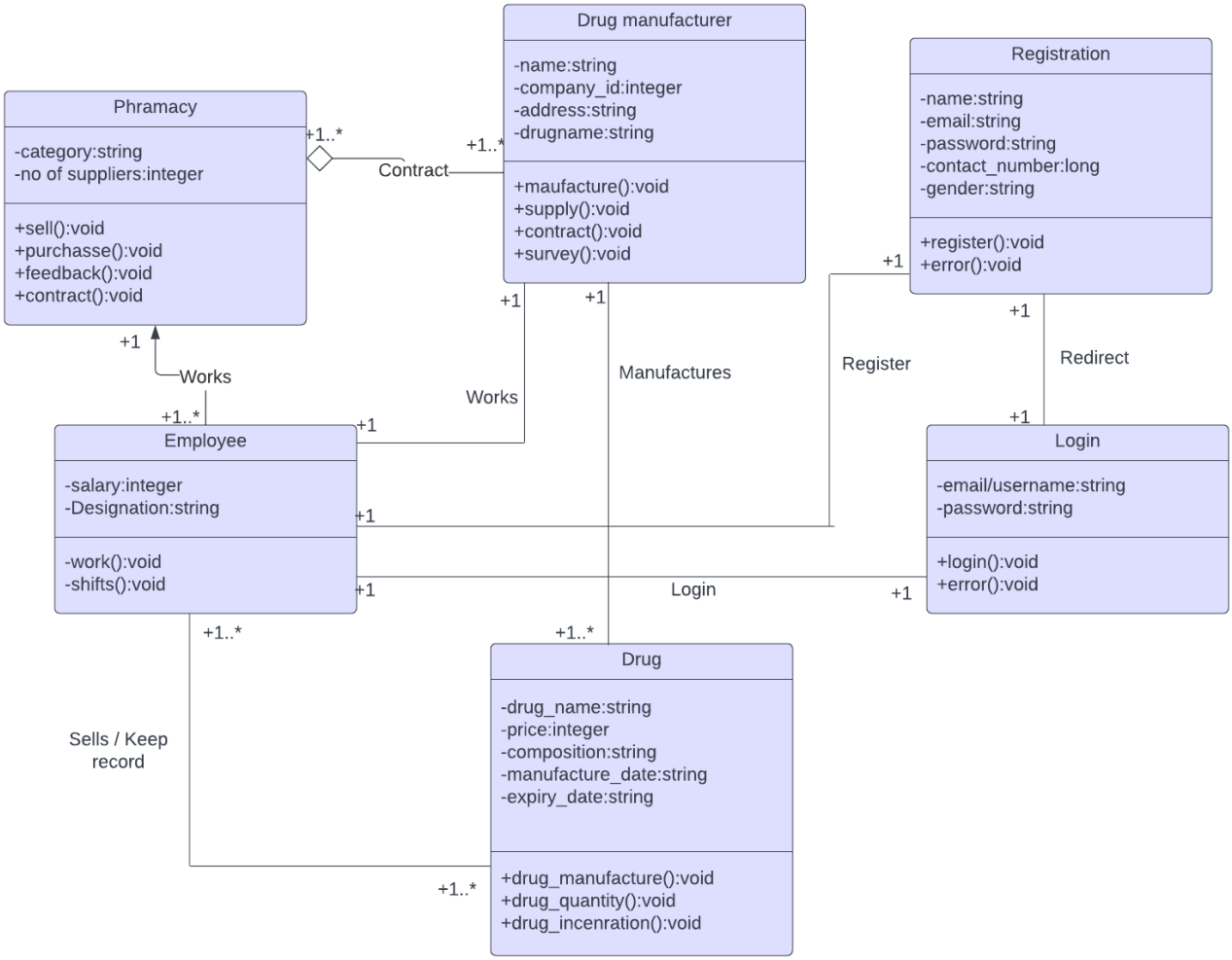
Experiment 4: Design a function-oriented diagram for the selected system

Solution:



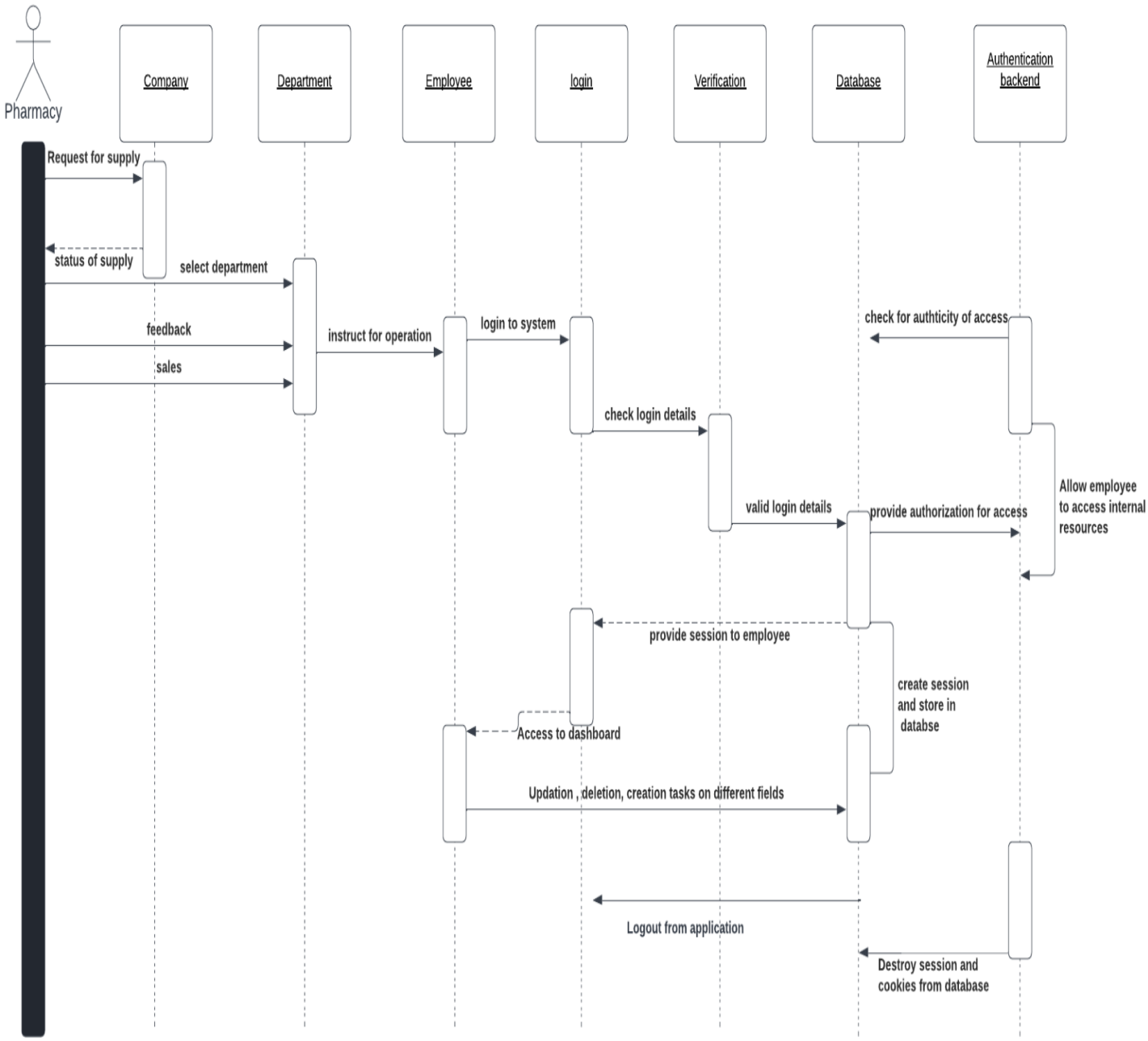
Experiment 5: Design a structural view diagram for the selected system.

Solution:



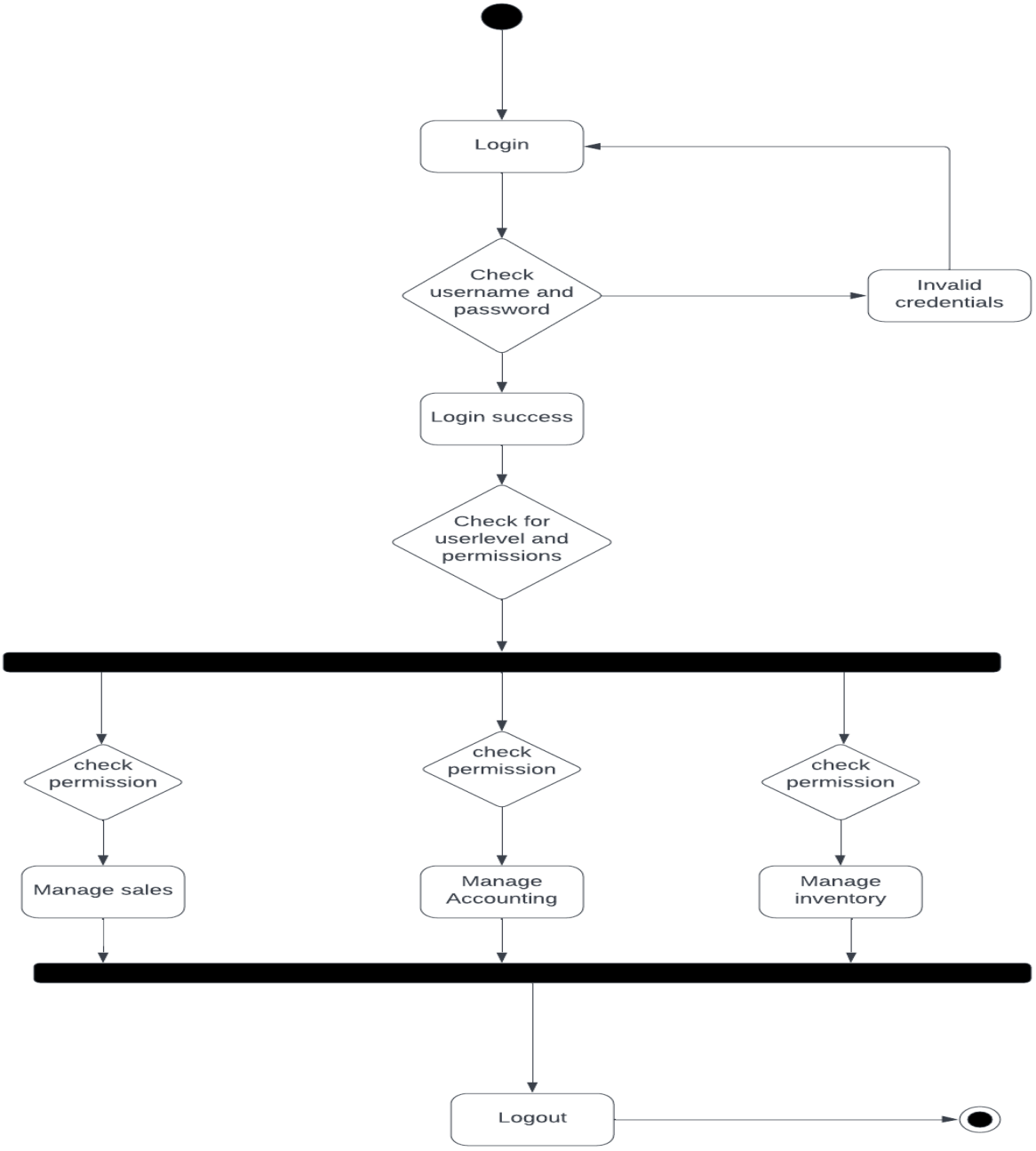
Experiment 6: Design the behavioral view diagram for the selected system

Solution:



Experiment 7: Design the behavioral view diagram for the selected system.

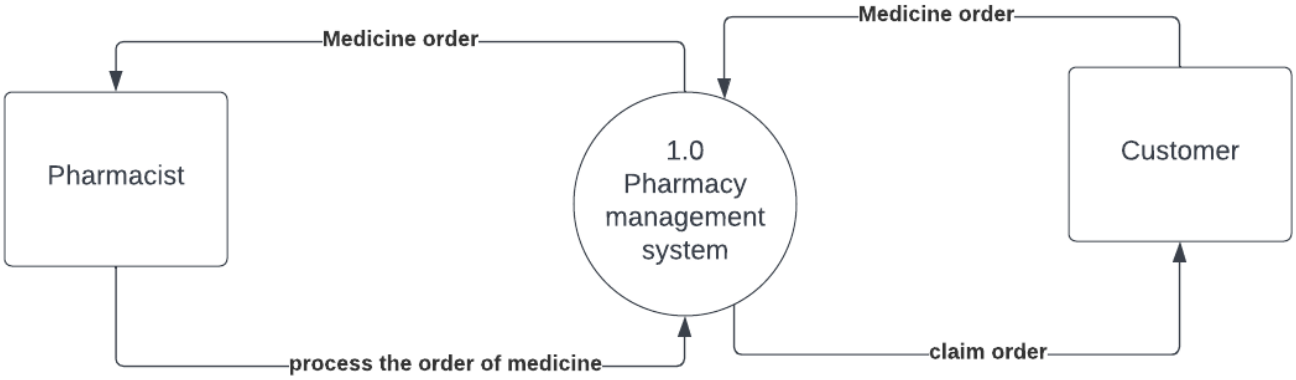
Solution:



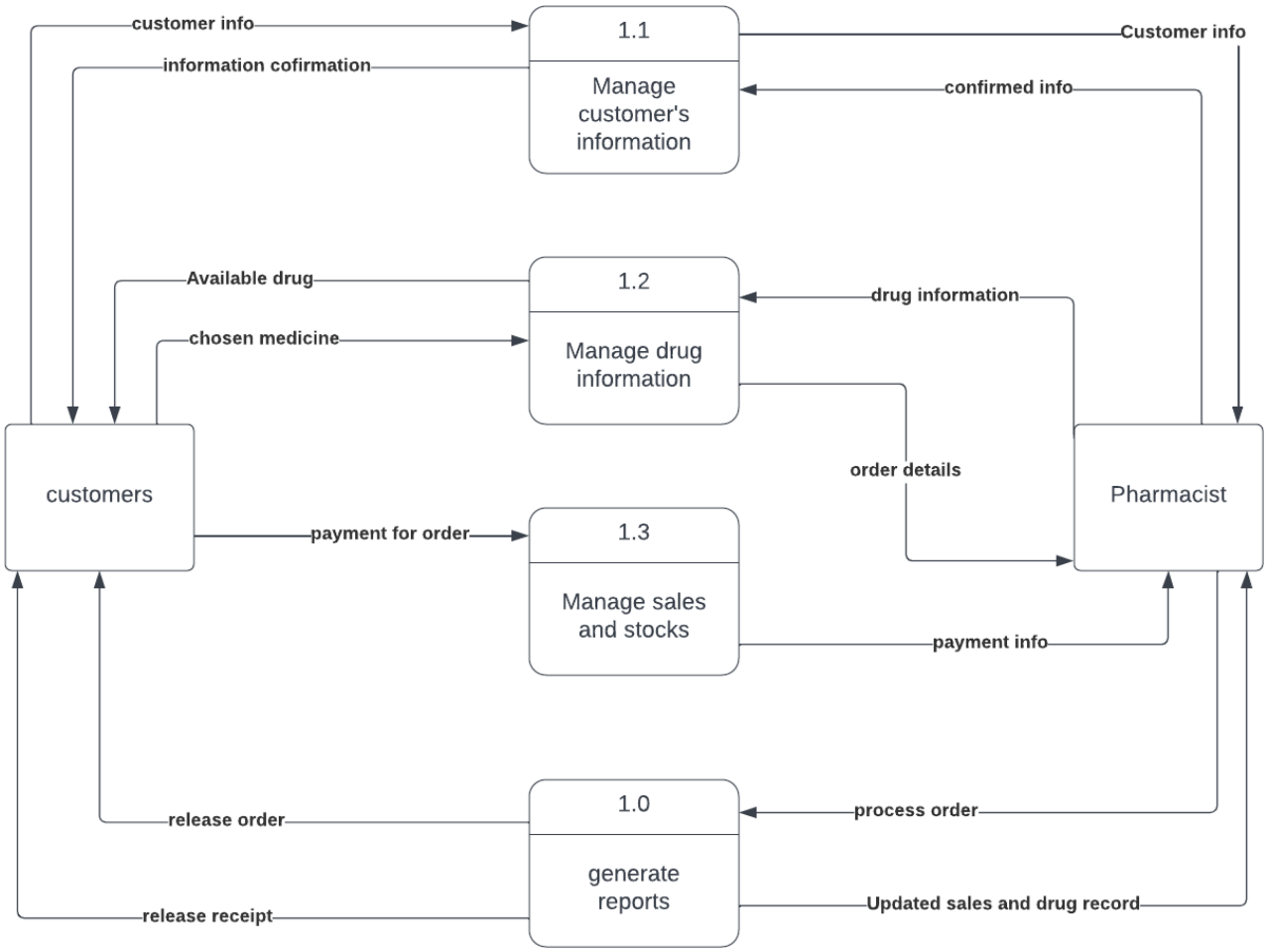
Experiment 8: Develop a Structured design for the DFD model for the selected system.

Solution:

DFD - Level 0



DFD - Level 1:


customer info
information cofirmation
1.1
Manage customer's information
Customer info
confirmed info
Available drug
chosen medicine
1.2
Manage drug information
drug information
order details
customers
Pharmacist
payment for order
1.3
Manage sales and stocks
payment info
release order
1.0
generate reports
process order
release receipt
Updated sales and drug record

Experiment 9: Implementation Planning and details for Selected system.

Solution:

Implementation Environment (Single vs Multi-user, GUI vs Non-GUI)

User friendly because it is so easy to use and it is including explanations for everything
User can access his login by more way
- Website
-Webapp

Program/Modules Specification:
1. Patient management system.
2. Doctor management system.
3. Drug management system.
4. Administrative rights management system.
5. Online appointment management system.
6. Invoice management system.
7. Medical services system.
8. Doctor services report system

Security Features:

1. FireWalls
Rely on trained personnel to set up firewalls to protect hospital systems from external
attacks. Firewalls can be based upon hardware or software depending on the size of the
hospital system.

2. Install Antivirus
Anti-virus software guards against malicious code that can compromise hospital systems. Viruses are a common way for hackers to gain access to hospital systems and are constantly evolving. Anti-virus updates are essential to security.

3. BackUp Data
Hospital systems are vulnerable to disasters — fire, flood, or hurricanes for example —in addition to malicious hacking. Backing up the data on a regular basis and keeping those backups securely stored are key to recovering from disaster.

4- Authenticate Every User
Determine who needs access to internal records and allow only them to access that data. Use audits to verify who is accessing what data. Don't neglect to remove employee access of those who have been terminated.

5. Strong Password Policy
Use secure password best practices such as

1. Use multiple passwords for different systems
2. Change passwords regularly
3. Use complicated passwords that aren't easily guessed
4. Use multi-factor authentication (which is required for e-prescribing systems)
5. Allow password resets for users who forget their passwords to prevent written
6. passwords left in vulnerable locations

6. Restrict Physical Access
Missing laptops or data storage devices are common sources of pharmacy data breaches. These devices need to be kept in secure locations where unauthorized users cannot access them. Security policies should restrict their use outside of facilities to prevent loss.

7. Encrypt Data
Encrypt healthcare data that is being transferred or stored to make it difficult (ideally, impossible) for hackers to interpret data even in the event of a successful breach.
Decisions need to be made concerning what data is to be encrypted and by which methods will be used.

Experiment 10: Do Testing (choose appropriate testing strategy or techniques suitable to your system)

Solution:

Testing Plan

Testing is an important phase in software development. Here for our software, we need to test if the login of the owner, employee, or customer is correct or not. Whether or not the employee with authorized access is able to make changes to the database or not. If the backend is working properly and the authenticity of each user is being checked also the activities done by the user are being checked throughout the session. If there is any security misconfiguration. In our testing phase for the pharmacy management system, we can take OWASP Top 10 as our guideline to test the software for different bugs and misconfigurations.

Testing Methods

1. System integration testing:

making sure that all different parts of the application are able to communicate with each other.

2. User Acceptance testing:

test performed by the user/client to see the application is built as required

3. Black-Box Testing: The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

5. White-Box Testing: White-box testing is the detailed investigation of the internal logic and structure of the code. White-box testing is also called glass

testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

Test Samples/Cases:

Name of test	Registration
Item/Feature being tested	Whether the system is able to create the user Account
Sample input	Allow user to register on providing specified Information.
The expected output	User account should be created.
The actual output	User account created successfully or Error Box if a user is not providing correct info.
Remarks	Module is working properly.
Pass/Fail?	Pass/Fail

Name of test	Sales Data Manipulation
Item/Feature being tested	If or not the employee with authorized access is able to alter the data in the sales table

Sample input	Try altering data in the sales table using an employee account.
The expected output	Alteration in data in the sales table.
The actual output	Alteration in data is seen
Remarks	Module is working properly.
Pass/Fail?	Pass

Name of test	Purchase
Item/Feature being tested	Whether the pharmacy owner is able to purchase medicine from the contracted company.
Sample input	Owner of the pharmacy makes a purchase order
The expected output	Order should be placed.
The actual output	Owner was able to place an order with the company.
Remarks	Module is working properly.
Pass/Fail?	Pass

Name of test	Sell
Item/Feature being tested	Whether the customers are able to buy the medicines or not
Sample input	create a customer account and try to buy medicine.
The expected output	Medicine order should be placed
The actual output	Customer was able to place the order . .
Remarks	Module is working properly.
Pass/Fail?	Pass

Name of test	Employee data manipulation
Item/Feature being tested	Whether the owner account is able to manipulate the employee data or not in the employee table.
Sample input	Owner adds new employee details.
The expected output	New employee details should be added to the employee table in the database.
The actual output	Employee data was successfully added.

Remarks	Module is working properly.
Pass/Fail?	Pass

Experiment 11: Specify the Limitation, Future Enhancement, and Conclusion for the selected system.

Solution:

Limitation:
The current system doesn't check the dose of the drug it's selling, the employee has the full authorization to update or edit any data and hence can cause the problem of permission conflict. The software has the ability to store financial information but it is not secure with the state of the art security system.

Future Enhancement:
There is a lot of room for improvement in the pharmacy management system. Here we can also use ML algorithms to help suggest the customers buy medicines that are best when suffering from certain diseases. ML can also be used to see the current data of diseases that are occurring more by seeing the number of medicines purchased for that disease. We can take care of the drug addicts buying drugs online by, making sure of complete validation of customers with restricted medicine demand using id cards, citizenships, etc.

Conclusion:
The pharmacy management system is useful for all kinds of pharmacy stores. It is very efficient in keeping track of the drugs, sales, quantity, employees, feedback and purchase and sale financials. The system plays a vital role in operating the store smoothly with less manpower and less overhead for the owner. The system

can use ML in future to automate, suggest and predict the sales and drugs types depending on the geography and type of patients around the place.

Experiment 12: Introduction to GIT and account creation on GIT.

Introduction:

Git:

Git is a version control system that lets us track changes we make to our files over time. With Git, you can revert to various states of our files. We can also make a copy of our file, make changes to that copy, and then merge these changes to the original copy.

How to install Git:

In order to use Git, we have to install it on our computer. To do this, we can download the latest version on the official website. We can download git for our operating system from the options given. Mine is windows so I have downloaded it for windows.

After that go to the downloads folder and double click the Git.exe file. Just press on next, next, next, I agree and we have downloaded Git successfully. Now let's configure git to connect to its online hosting repo GITHUB with the necessary credentials.

How to configure Git:

To configure our Git account in Windows, open up a command prompt and follow the instructions below:

To set a username, type and execute these commands:

git config --global user. name "YOUR_USERNAME" and

git config --global user.email "YOUR_EMAIL".

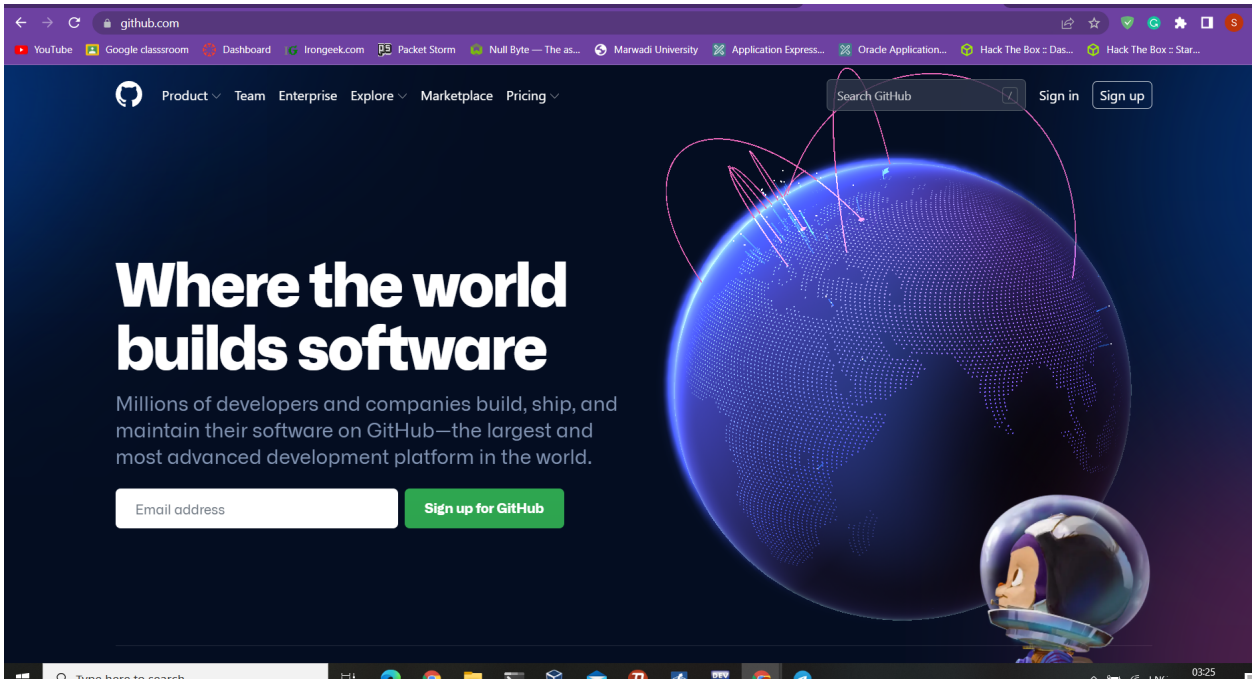
Just make sure to replace "YOUR_USERNAME" and "YOUR_EMAIL" with the values we choose.

Account Creation on Git:

We actually don't need an account to use git, but to host files on a remote server we do need a Github account. Github is a repo hosting service and we can make an account similar to any other website.

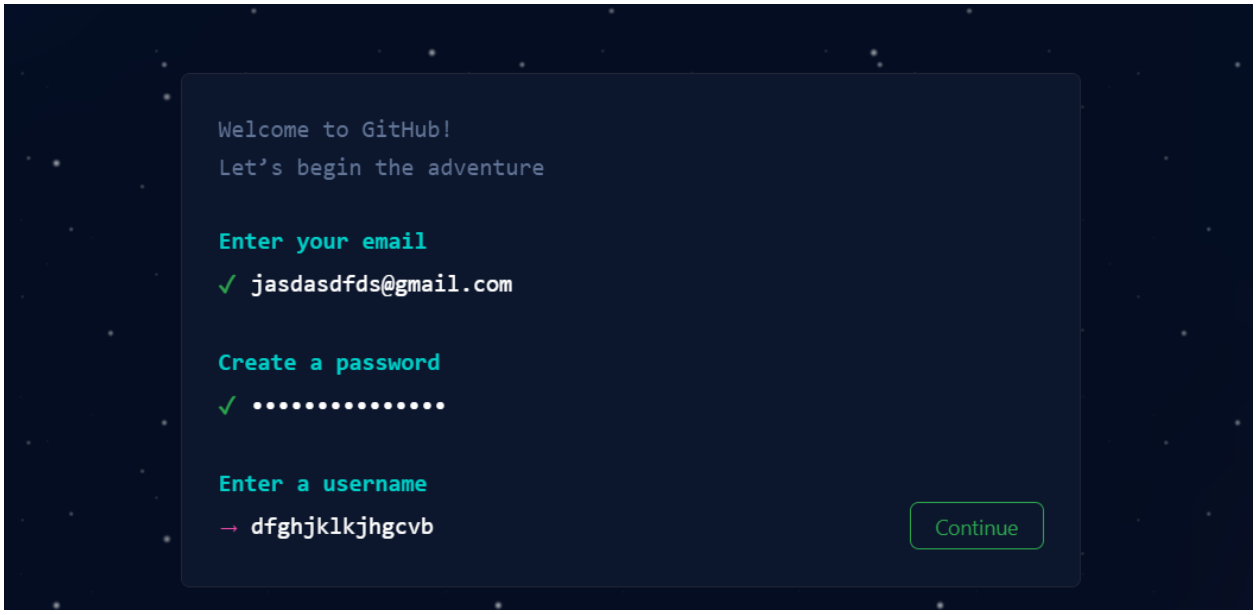
Steps:

1. Go to https://github.com/join.



2. Choose to **Sign up for GitHub**, and then follow the instructions.
3. Type a user name, your email address, and a password.

   (make sure those are the same as that of the git config file.)



Profile link : https://github.com/InfooGeek

Experiment 13: Introduction to Team Foundation server tool.

Introduction:

Microsoft developed a Team Foundation Server or TFS to manage the teams and the way they work. It is basically a management tool used in project management, reporting, requirements gathering, and testing. It actually covers the entire software development life cycle and operates in Microsoft Windows. It consists of version control, issue-resolving, and application management. This provides end-to-end development of software and testing. This is a back-end management system and uses Git to control the source code. It represents the automation process and testing of an application. It is provided with visual studio code.

Team Foundation Server (TFS) offers both web-based and client-based solutions for test management:

- The Test hub in the TFS web portal, as a web-based solution, can work across all platforms and with all browsers.
- Microsoft Test Manager, as a client solution, requires Build System, Test Controller, and Test Agent to communicate with TFS.

Experiment 14: Study of Various Testing Tool:

i) Win Runner 8.0 - Checkpoints in Win runner, Data-Driven and Batch Testing.

i) Win Runner 8.0

1. **Checkpoints in Win runner:**

Checkpoints allow you to compare the current behavior of theapplication being tested to its behavior in an earlier version.

We can add four types of checkpoints to your test scripts:

1 GUI checkpoints verify information about GUI objects. For example, you can check that a button is enabled or see which item is

selected in a list. See, “Checking GUI Objects,” for more information.

2. Bitmap checkpoints take a “snapshot” of a window or area of your application and compare this to an image captured in an earlier version. See, “Checking Bitmaps,” for more information.

3. Text checkpoints read the text in GUI objects and in bitmaps and enable you to verify their contents. See, “Checking Text,” for more information.

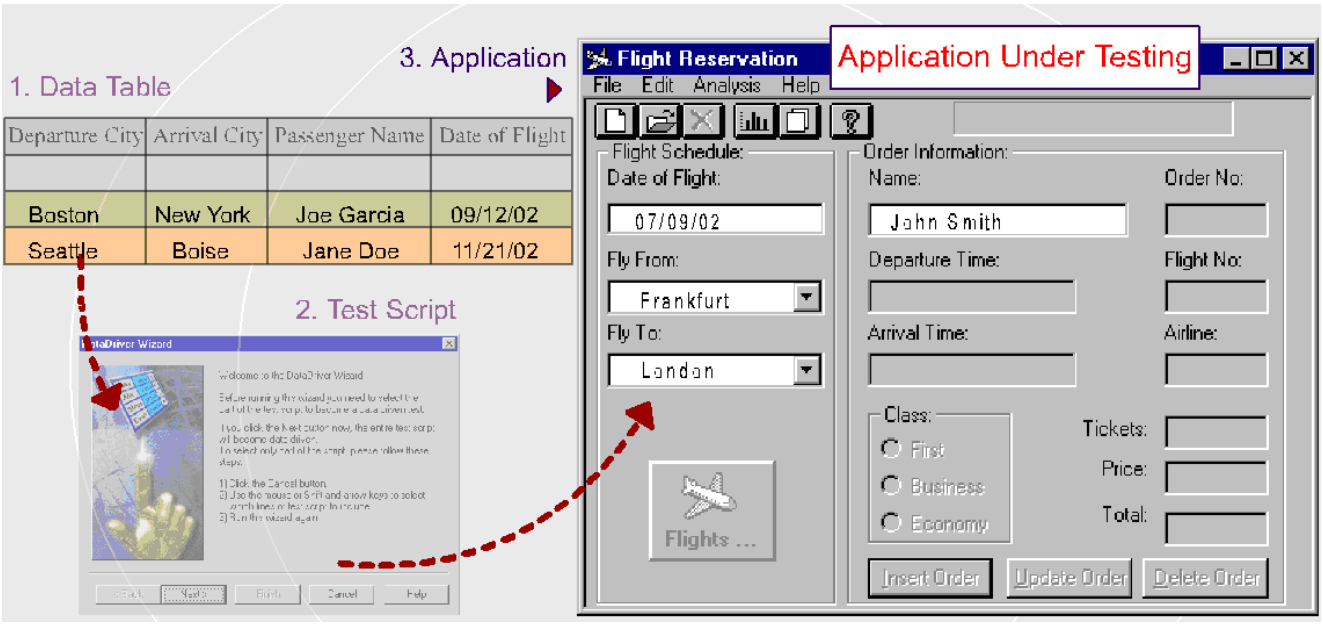
4. Database checkpoints check the contents and the number of rows and columns of a result set, which is based on a query you create on your database. See, “Checking Databases,” for more information.

Data-Driven Testing:

When you want to test your application, you may want to check how it performs the same operation with multiple sets of data. So we use Data-Driven Test. By replacing the fixed values in your test with values stored in a data table (an external file), you can generate multiple test scenarios using the same test.

Two ways of Data-Driven Test

1. Data-driven Wizard
2. Modify Test Script manually



Batch Testing:

A Batch test is a test script that contains call statements to other tests

1. It opens and executes each test and saves the test results.
2. It suppresses the error message that occurs while running the test script
3. A test becomes a batch test when you select the run in batch mode option

e.g.

GUI_load(“a1.

gui”); call

“a1”()