

9. Screen Layout

Login Page :

The image shows a browser window with a title bar containing standard icons (minimize, maximize, close) and the text 'x'. The main content area of the browser displays the text 'A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY' in red. Below this, there is a large rectangular box labeled 'Login page' in blue. Inside the 'Login page' box, there are two input fields: 'User name' with a user icon on the right, and 'Password' with a padlock icon on the right. Below these fields is a 'submit' button. The browser window also features a vertical scrollbar on the right side.

New User Registration:

—

X

A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY

MY DETAILS

VIEW FILE

UPLOAD FILE

DOWNLOAD
FILE

FEEDBACK

LOGOUT

↑

FIRST NAME	
LAST NAME	
DATE OF BIRTH	
AGE	
USER ID	
PASSWORD	
MOBILE NO	
EMAIL ID	
GENDER	

SUBMIT

Upload File Page:

—

X

A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY

↑

MY DETAILS	VIEW FILE	UPLOAD FILE	DOWNLOAD FILE	FEEDBACK	LOGOUT
------------	-----------	-------------	---------------	----------	--------

↑

NAME	
TOPIC	
DOCUMENT	<div>CHOOSE FILE</div>
DATE	
REQUEST	

SUBMIT

↓

View File Page:

X

A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY

MY DETAILS

VIEW FILE

UPLOAD FILE

DOWNLOAD FILE

FEEDBACK

LOGOUT

↑

TOPIC	DOCUMENT	REQUEST
.....
.....
.....

↑

Download Page :

X

A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY

MY DETAILSVIEW FILEUPLOAD FILEDOWNLOAD FILEFEEDBACKLOGOUT

NAME	TOPIC	DOCUMENT	STATUS
*****	*****	*****	REJECTED
*****	*****	*****	DOWNLOAD
*****	*****	*****	REJECTED

FEEDBACK PAGE : This page describes about user feedback and details .

A DATA ANALYTICS APPROACH TO THE CYBERCRIME UNDERGROUND ECONOMY													
MY DETAILS	VIEW FILE	UPLOAD FILE	DOWNLOAD FILE	FEEDBACK	LOGOUT								
<div>↑</div> <table border="1"><tbody><tr><td>NAME</td><td></td></tr><tr><td>MOBILE NUMBER</td><td></td></tr><tr><td>EMAIL ID</td><td></td></tr><tr><td>FEEDBACK</td><td></td></tr></tbody></table> <div>SUBMIT</div>						NAME		MOBILE NUMBER		EMAIL ID		FEEDBACK	
NAME													
MOBILE NUMBER													
EMAIL ID													
FEEDBACK													

10 Database Design

10.1 ER Diagram

An Entity-Relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure. ER Diagrams are most often used to design or debug relational databases.

User:

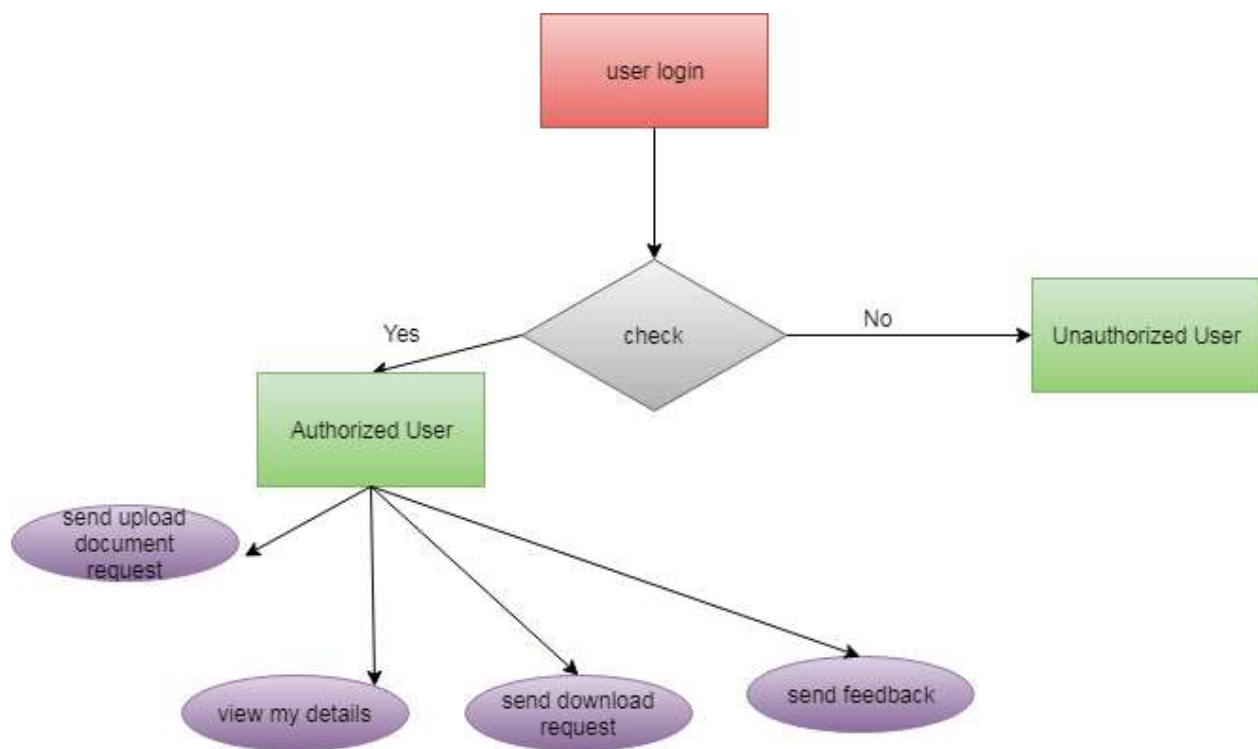


Figure:ER diagram for user module

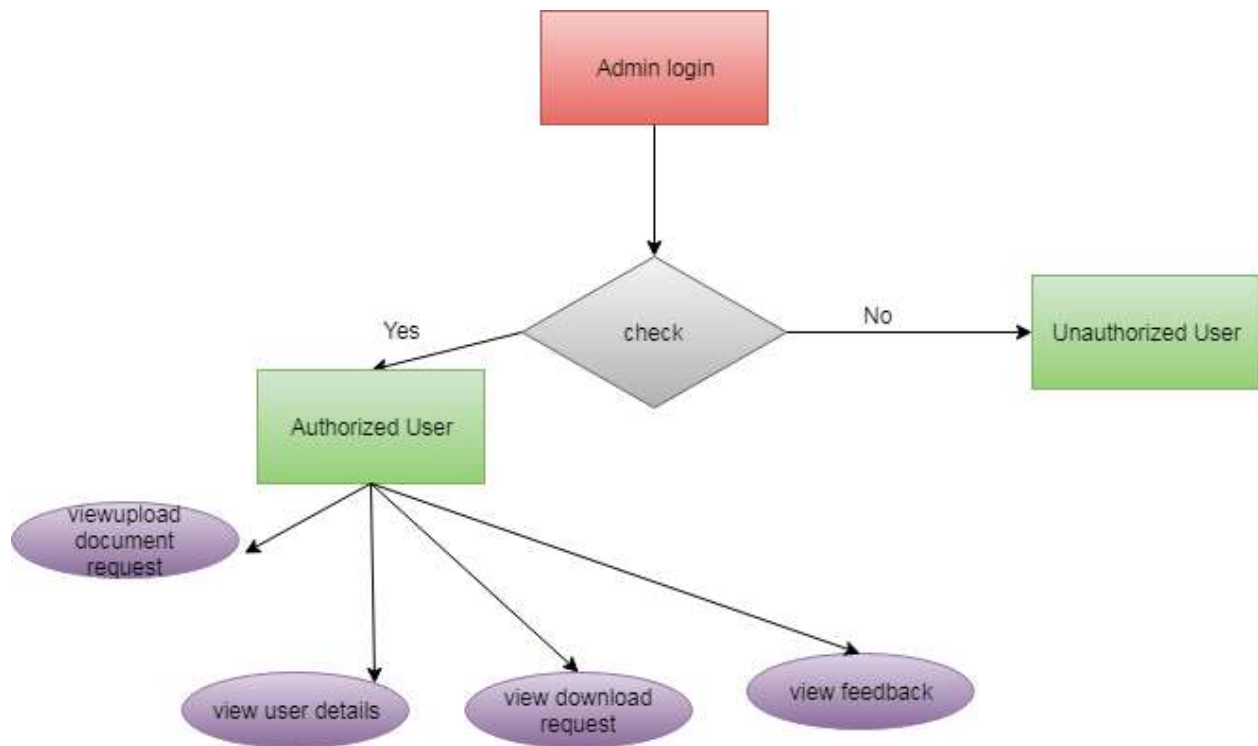
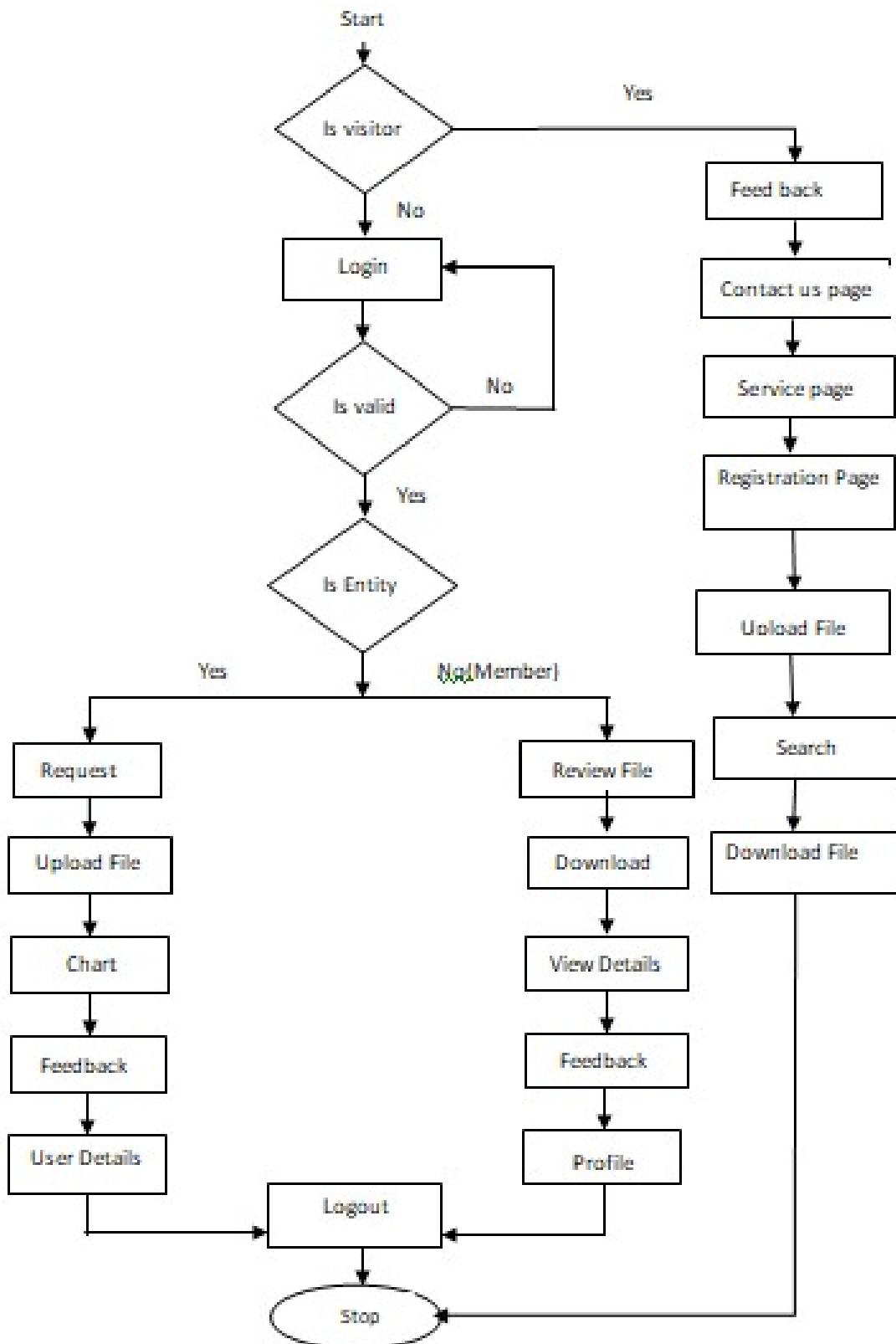
Admin:

Figure:ER diagram for admin module

Flow Chart:



10.2 Data Flow Diagram

Data-flow diagram is a way of representing a flow of a data of a process or a system and information about the outputs and inputs of each entity and the process itself.

Business Entity/Admin :

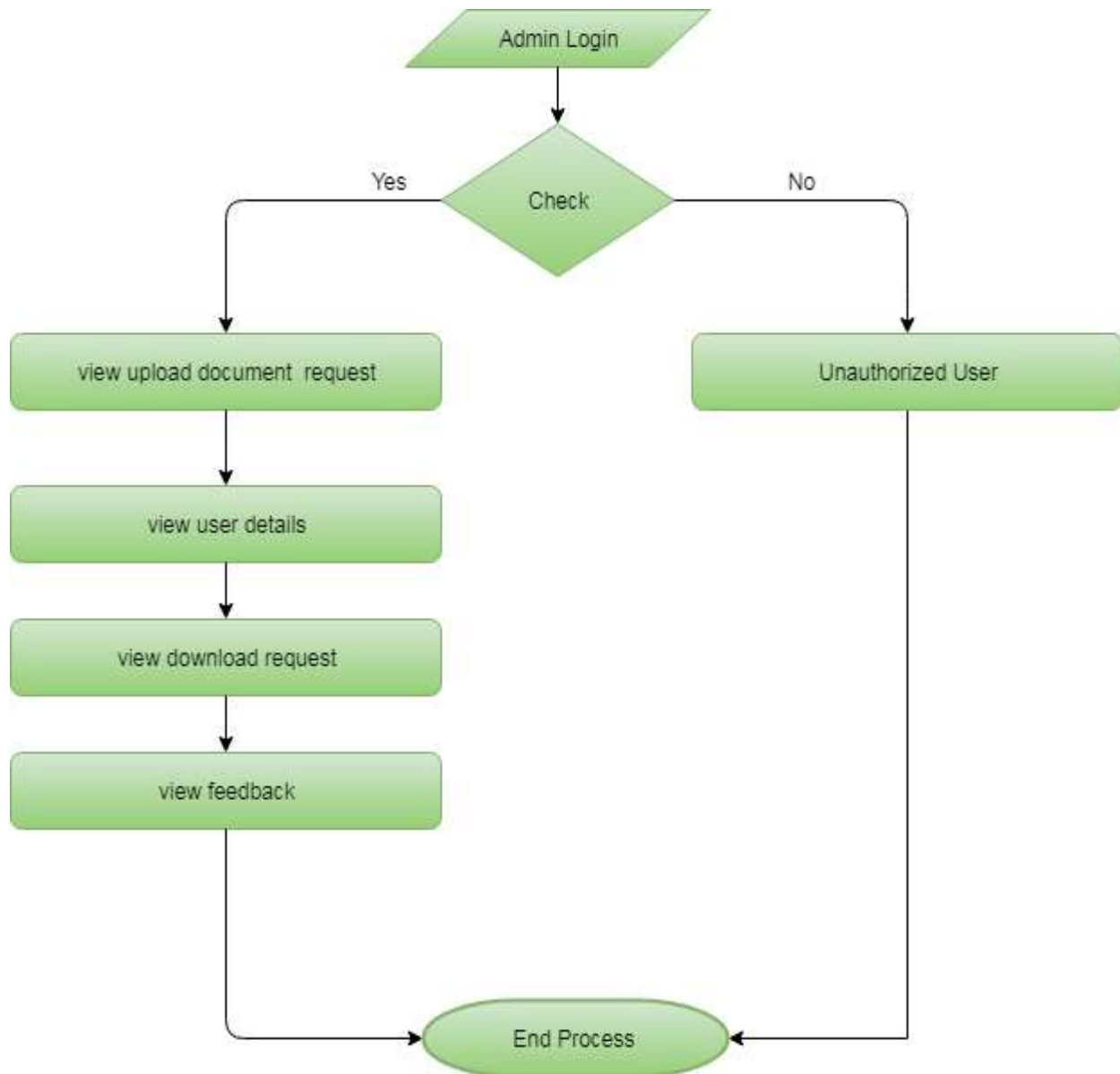


Figure:Data Flow diagram for admin module

User:

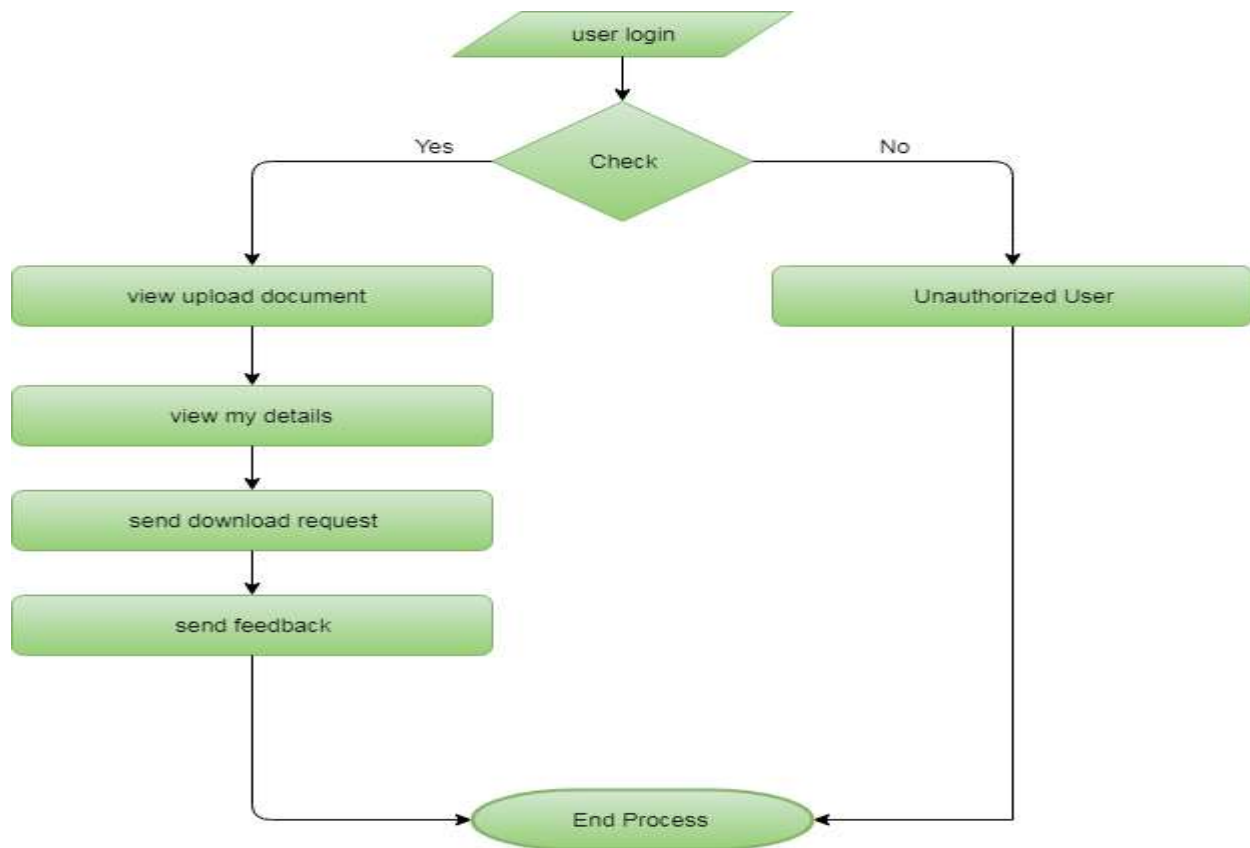


Figure:Data Flow Diagram for user

11 Table Struture

Table Name: RegisterModel

Column	Type	Default
<u>id</u>	int(11)	-
firstname	varchar(100)	NULL
lastname	varchar(100)	NULL
dob	varchar(100)	NULL
age	varchar(100)	NULL
userid	varchar(100)	NULL
password	int(11)	NULL
mobilenumber	varchar(100)	NULL
emailid	varchar(100)	NULL
gender	varchar(100)	NULL

Table Name: UploadModel

Column	Type	Default
<u>id</u>	int(11)	-
name	varchar(100)	NULL
topic	varchar(100)	NULL

document	varchar(100)	NULL
date	varchar(100)	NULL
request	varchar(100)	NULL
userDet_id	varchar(100)	NULL

Table Name: ChatModel

Column	Type	Default
<u>id</u>	int(11)	-
senderId	int(11)	NULL
chat	varchar(300)	NULL
userId_id	int(11)	NULL

Table Name: RequestModel

Column	Type	Default
<u>id</u>	int(11)	-
accessone_id	int(11)	NULL
accesstwo_id	int(11)	NULL

request	varchar(200)	NULL
---------	--------------	------

Table Name: FeedbackModel

Column	Type	Default
<u>id</u>	int(11)	-
name	varchar(300)	NULL
mobilenumber	varchar(300)	NULL
emailid	varchar(300)	NULL
feedback	varchar(300)	NULL

Table Name: UserChat

Column	Type	Default
<u>id</u>	int(11)	-
cusid	varchar(200)	NULL
spkid	varchar(200)	NULL
userchats	varchar(200)	NULL

Table Name: Algorithm_Model

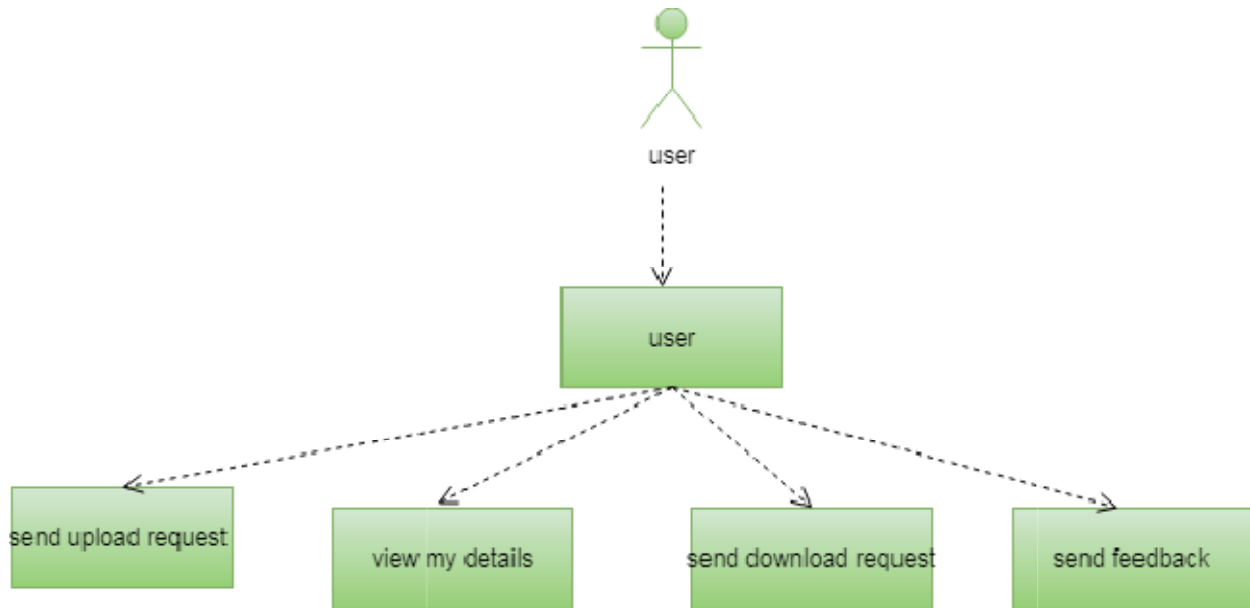
Column	Type	Default
<u>id</u>	int(11)	-

algorithm_name	varchar(100)	NULL
precisions	varchar(100)	NULL
Recall	varchar(100)	NULL
Accuracy	varchar(100)	NULL
True_Negative_Rate	varchar(100)	NULL

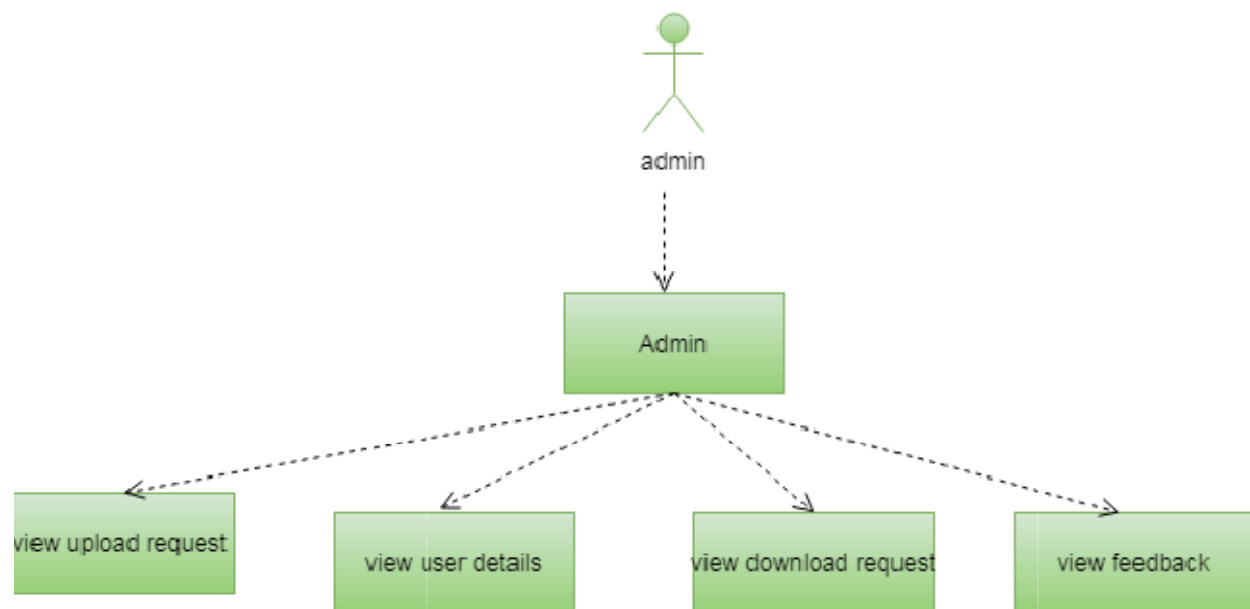
12. Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

Business Entity/User:



Admin:



13. Sequence diagram

a) User

When Admin sign up successfully he/she can view, add and delete eco products category and also he/she can view ,add and delete any eco products, as well as view and respond to customer orders and can also view customer feedback for future upgradation and at last admin also have an option to update his/her password.

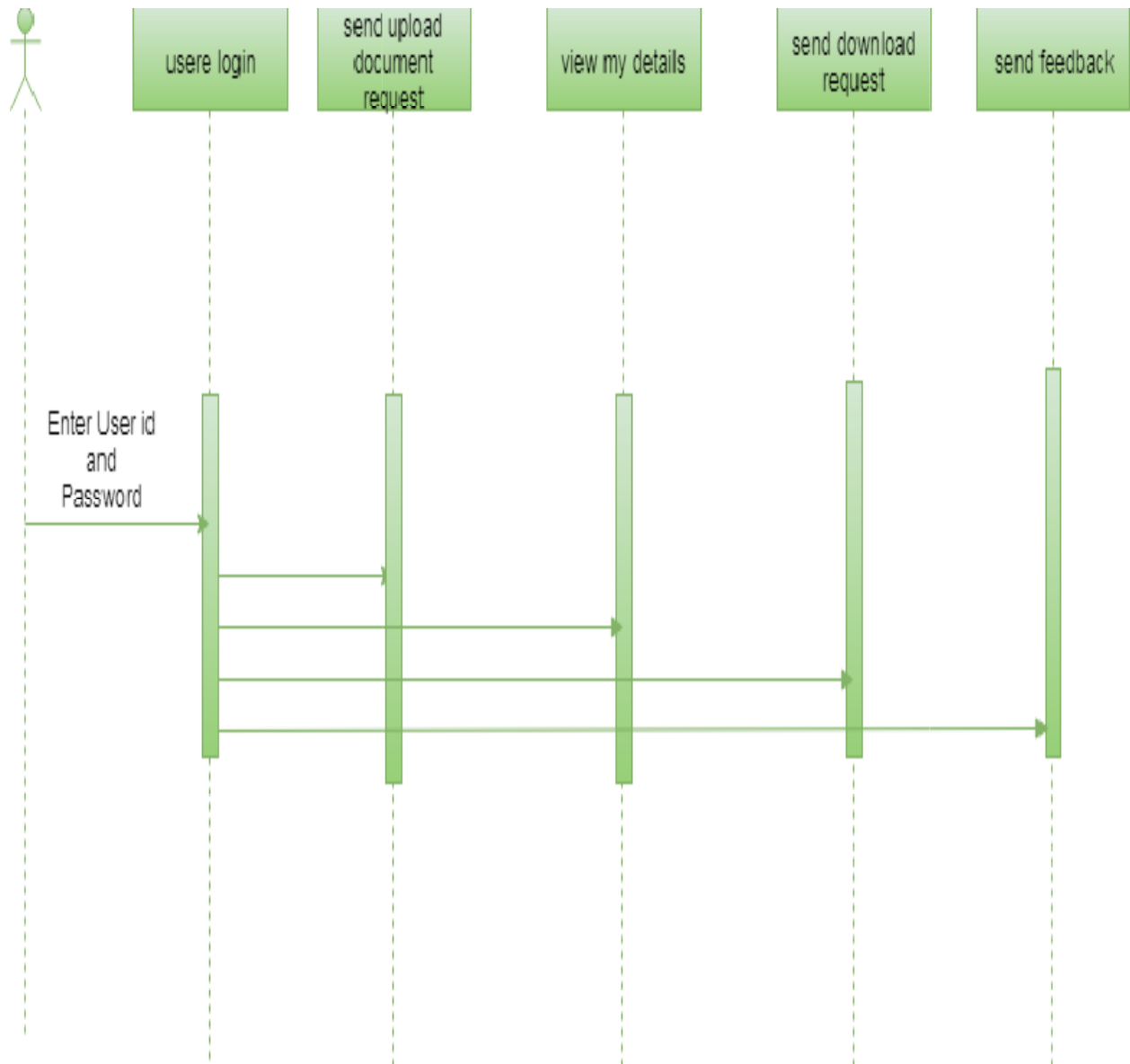
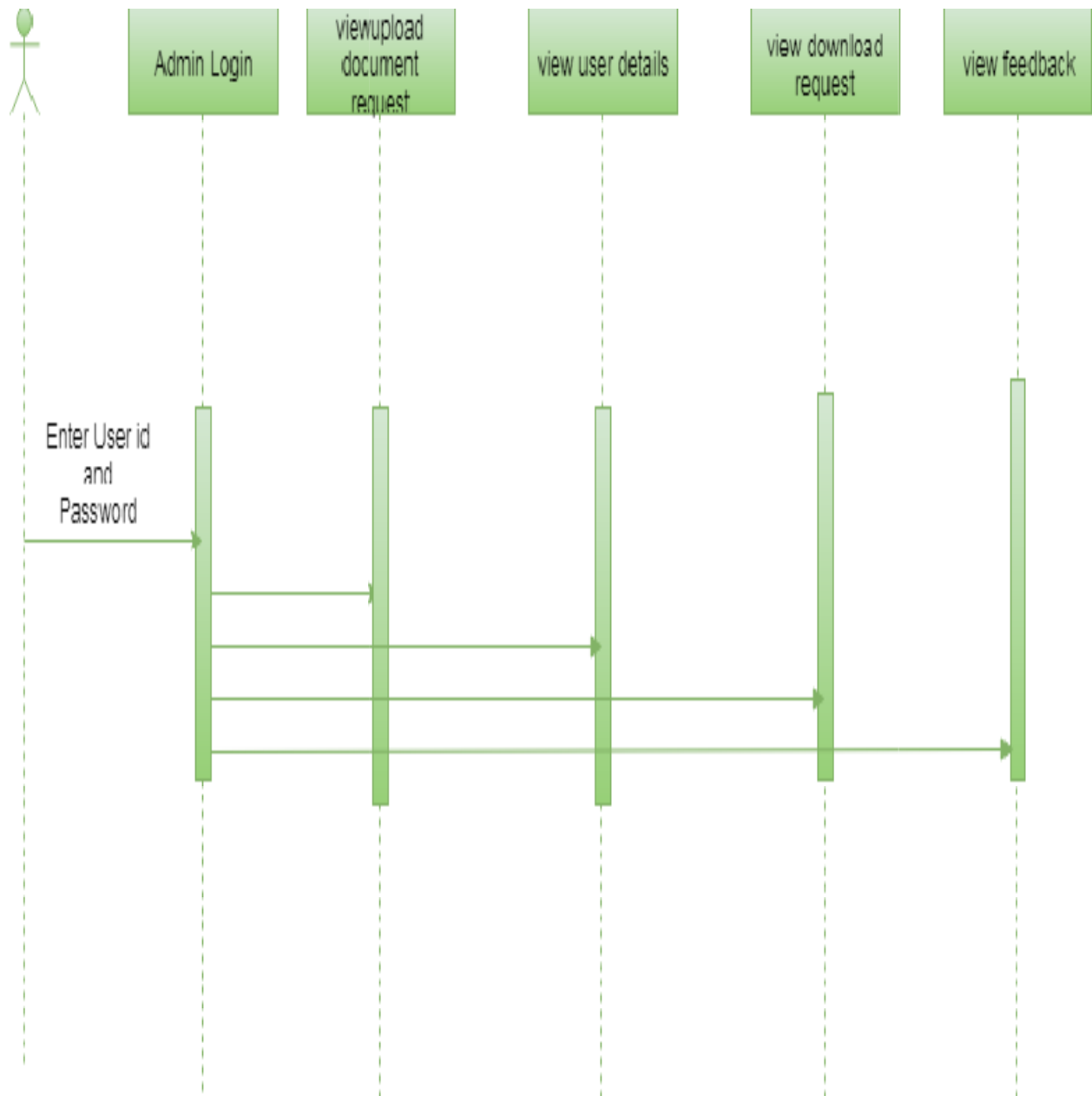


Figure: Sequence Diagram For User

b) Admin:

**Figure:** Sequence Diagram For Admin

When a user gets signed up, he/she can view or delete review order and can add to place a order , also can view order history and can add his/her feedback and at last he/she can also update his/her profile

14. Class Diagram

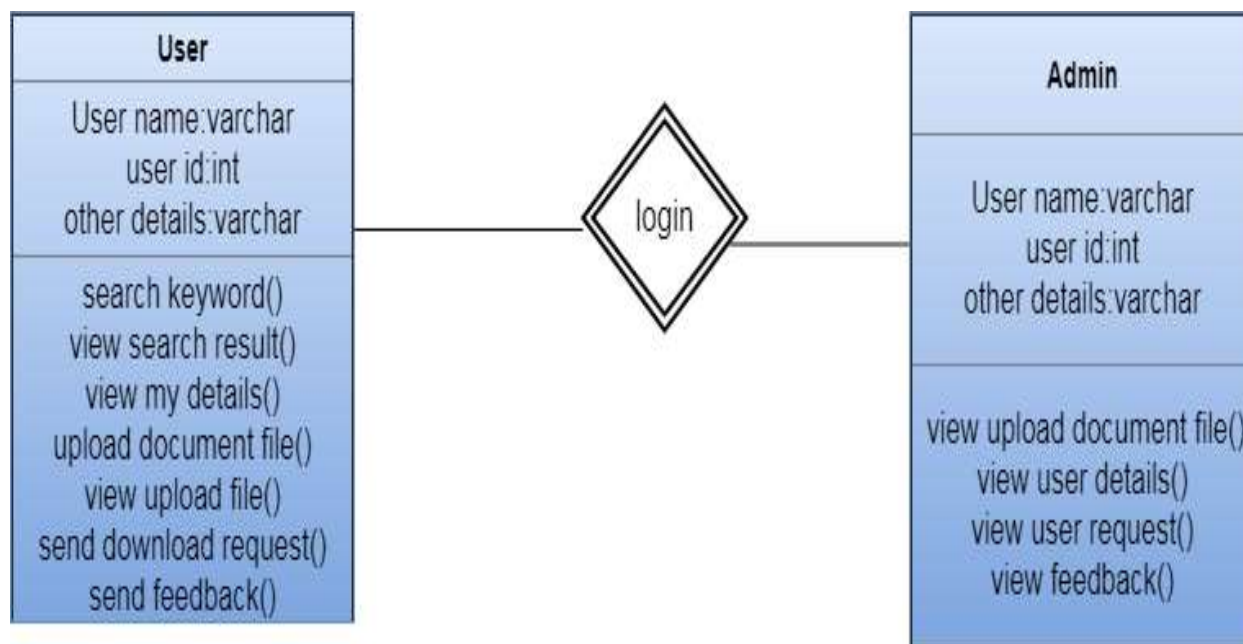


Figure:Class Diagram

In software engineering, a **class diagram** in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.^[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

15. SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered

Test case ID	Description	Test steps	Expected value	Actual value	OK/Err or
1.	Verify login page	Input username and password	Login page	Invalid data	error
2	Verify login page	Input username and password	Login page	Login page	ok
3.	Verify registration page	Registration	User profile	Registration failed	error
4.	Verify registration page	Registration	User profile	User profile	ok
5.	Query is to be posted	Posting query	Query is posted	Unable to post the query	error
6.	Query is to be posted	Posting query	Query is posted	Successfully posted	ok
7.	Files to be upload	Upload file	File to upload	File name is entered	ok
8.	Files to be upload	Upload file	File to upload	Unknown file	error
9.	Files to be requested	Enter request file	File to be send	File name is Request	ok
12.	Files to be reuested	Enter request file	File to be send	File not found	error
13.	Chat Module	Enter text	Negative Text	Possitive	error

			is entered	Text is entered	
14.	Chat Module	Enter text	Possitive Text is entered	Possitive Text is entered	ok
15	FeedBack	Text	Ok

16. Limitations/Constraints/Drawbacks

- Cybercrime has undergone a revolutionary change, going from being product-oriented to service-oriented. The cybercrime underground has a highly professional business model that supports its own underground economy. This business model, known as CaaS, is “a business model.
- CaaS is referred to as a do-it-for-me service, unlike crimeware which is a do-it-yourself products.
- It is not secured process.
- Over under traction is invents.
- Download files, time is invited.
- The cyber-crime differs from general crime in many ways; we need to conduct a variety of analyses using a large data set.
- A previous study proposed a data mining framework for crime, dividing crimes harmful to the general public into eight categories:
 - 1. Traffic violations.
 - 2. Sex Crime.
 - 3. Theft.
 - 4. Fraud.
 - 5. Gang/Drug Offenses.
 - 6. Arson.
 - 7. Violent crime.
 - 8. Cyber-crime.
- Although this previous study explained how data mining techniques could be applied to crime analysis, it did not consider the specific features of cyber-crime.
- Consequently, the CaaS business model can involve the following roles:
 - i. writing a hacking program
 - ii. performing an attack
 - iii. commissioning an attack
 - iv. providing an attack server (infrastructure) and
 - v. laundering the proceeds.
- Sood and Enbody have suggested that crimeware marketplaces have three key elements, namely actors (e.g., coders, operators, or buyers), value chains, and modes of operation.
- Periodic monitoring & analysis of the content of cybercrime marketplaces could help predict future cyber threats.