

# ***CAMPUS CENTRIC SECOND HAND EXCHANGE***

Tadikonda Tejaswi

*Department of Computer Science and Engineering*

*Shri Vishnu Engineering College for Women(A)*

Bhimavaram, India

Bathula Gowthami

*Department of Computer Science and Engineering*

*Shri Vishnu Engineering College for Women(A)*

Bhimavaram, India

Veluguri Lakshmi Sindhu

*Department of Computer Science and Engineering*

*Shri Vishnu Engineering College for Women(A)*

Bhimavaram, India

Mohammed Afreen

*Department of Computer Science and Engineering*

*Shri Vishnu Engineering College for Women(A)*

Bhimavaram, India

Sunnapuram Sai Durga

*Department of Computer Science and Engineering*

*Shri Vishnu Engineering College for Women(A)*

Bhimavaram, India

***Abstract-*** In light of the circumstances, people are now calling for a green civilization, with resource conservation as a top priority. In today's world, technology has a significant impact on our daily life. The current generation enjoys the great convenience that sophisticated technology brings to us. Using it to deal with real problems is what we ought to do. It serves as an incentive for creating a professional on-campus website for trading used products. This project aims to create and implement a second-hand trading website for the college, providing a platform for students to effortless trading of old products. It not only saves students money but also helps to safeguard the environment. The influence that

advanced technology provided had motivated us to develop a professional on-campus website for trading old items.

***Keywords–*** resource conservation, platform, website, trading.

## **I. INTRODUCTION**

For international students or others, when first coming to the United State, everything is new. There are several problems as a beginner and

foreigner here. For example, renting an apartment, getting new furniture, buying new textbooks and so on. Since the culture and custom vary from other countries, the immigrant finds that these tasks are disasters. To start with, the new furniture or textbooks are very expensive, and it is even not worth buying a new one.

However, there are no known platforms, online or offline, which allow people to trade used items. Even searching the results online, the buyer still could not know whom to trust. Even if the buyer could find reliable websites later, transportation is still a big problem.

Therefore, this idea evolves, as a computer science project, to develop a platform focused on the CSUN. With the supervision of authorities from school, the safety issues could also be guaranteed.

According to situations that we are advocating green civilization, saving resources as much as we can should be a vital problem. There are over 3,900 international students on our campus currently that means students come and go constantly so that there might be giant old items that cannot be taken away. We all know that the books are expensive and could easily cost 1,000 dollars per year. A second-hand one could save students money.

Besides personal problem, nearly 50 percent of municipal waste generated in Finland in 2014 was burned. The percentage of burning has risen very fast. Ten years ago, just 12 percent of waste was burned. Disposal at landfill sites has, in turn, fallen more or less at the same rate. In practice, all municipal solid waste delivered for burning reduces the waste in landfill sites. Just under one-fifth of the total volume of waste was disposed at landfill sites. Municipal waste as an additional resource in energy production is running low.

This emphasis on the politics of sanitary reform and technologies of sewage treatment obscures the ways in which urban nature has been incorporated into technological waste-disposal strategies. Natural systems have been typically seen as outside urban networks: as a receiving environment that is subject to pollution by waste-waters, or as a water supply requiring purification by artificial methods such as filtration or chlorination.

As our country pushes forward the Internet, Internet industry flourishes, and online shopping

has become an indispensable part of people's life. Most young people have the experience of shopping on the Internet. It is because there are so swollen and broad market prospects that make more and more entrepreneurship people can get through online platform.

An environment-friendly society have been advocated; everyone should start from resource conservation. In school, especially many second-hand goods are thrown away in universities and colleges, which both damages the environment, and wastes of resources. For such rich second-hand resources, we can take reasonable way to handle it, build an Internet platform, make full use of the mature technology and social environment, let the students used resources can be reasonable shared and effectively used so as to make resources communication on campus, save cost, and make waste useful. So, this website creates campus second-hand trading platform is feasible and has certain practical value.

Existing websites do have fundamental customers. There are still some limitations. Since these platforms are open to the whole citizens, we cannot easily find what we need for school and the transportation is not convenient. We are trying to not move heavy items back to home. A trade application which just focuses on a single university is necessary.

Nowadays, advanced technology is greatly influencing our lives. People are taking smart phones everywhere for chatting with friends, paying for items, ordering take-out and so on. It has become a custom to university students. That provides motivations for developing a professional on campus application for trading old items.

The main goal of the application would be achieved to help the students deal with daily problems and satisfy them as most as possible.

## II. SYSTEM DESIGN

### A. Existing System

- Existing websites do have fundamental customers.
- Since these platforms are open to the whole citizens, we cannot easily find what we need for campus.

### B. Proposed System

- A trade website which just focuses on a single university.
- This website is a on campus massive potential market of second-handing trading to save money and also to help in protecting the environment.
- The students usually generate a great deal of spare items such as books, drafters, chairs, etc. can upload their items in this website and sell them.

### C. Feasibility Study

Feasibility Study in Software Engineering is a study to evaluate feasibility of proposed project or system. Feasibility study is one of stage among important four stages of Software Project Management Process. As name suggests feasibility study is the feasibility analysis or it is a measure of the software product in terms of how much beneficial product development will be for the organization in a practical point of view. Feasibility study is carried out based on many purposes to analyze whether software product will be right in terms of development, implantation, contribution of project to the organization etc.

- Technical Feasibility - Technical feasibility deals with the existing technology, software and hardware requirements for the proposed system. The proposed system “Campus Centric Second Hand Exchange” is planned to run on Java. Thus, the project is considered technically feasible for the development. The work for the project can be done with current equipment, existing software technology and available personnel. Hence the proposed system is technically feasible.
- Economic Feasibility - This method is most frequently used for evaluating the effectiveness of a Java. It is also called as benefit analysis. In this project “Campus Centric Second Hand Exchange” is developed on current equipment, existing software technology. Since the required hardware and software for developing the system is already available in the organization, it does not cost must developing the proposed system.
- Behavioral Feasibility - This project has been implemented by Java and it satisfies all conditions and norms of the organization and the users. This proposed system “Campus Centric Second Hand Exchange” has much

behavioral feasibility because users are provided with a better facility.

## III. SYSTEM REQUIREMENTS

### A. Software Requirements

- Tools Used:
  - Eclipse
  - Visual Studio Code
- Technologies Used:
  - Frontend:
    - Angular
    - HTML
    - CSS
    - Bootstrap
  - Backend:
    - Rest API
    - Hibernate
  - Database:
    - MySQL

### B. Hardware Requirements

- Operating System: Windows 8 and above
- RAM: 4GB and above
- Processor: Intel Core I3 and above
- Hard disk space: 500GB and above

## IV. SYSTEM DESIGN

### A. Introduction

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture. The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from problem domain to the solution domain. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The output of this phase is the design document. This document is similar to a blue print or plan for the solution, and is used later during implementation, testing and maintenance.

The design activity is often divided into two separate phase-system design and detailed design. System design, which is sometimes also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats, output formats, as well as the major modules in the system and their specifications are decided.

A design methodology is a systematic approach to creating a design by application of set of techniques and guidelines. Most methodologies focus on system design. The two basic principles used in any design methodology are problem partitioning and abstraction. A large system cannot be handled as a whole, and so for design it's partitioned into smaller systems. Abstraction is a concept related to problem partitioning. When partitioning is used during design, the design activity focuses on one part of the system at a time. Since the part being designed interacts with other parts of the system, a clear understanding of the interaction is essential for property designing the part.

### B. UML Diagrams

Unified Modeling Language (UML) is general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

Use of UML Diagrams:

- Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.
- Businessmen do not understand code. So, UML becomes essential to communicate with non-programmers' essential requirements, functionalities and processes of the system.
- A lot of time is saved down the line when teams are able to visualize processes, user interactions and static structure of the system.

UML is linked with object-oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:

- Structural Diagrams – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.
- Behavior Diagrams – Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

Object Oriented Concepts Used in UML -

- Class - A class defines the blue print i.e., structure and functions of an object.
- Objects - Objects help us to decompose large systems and help us to modularize our system. Modularity helps to divide our system into understandable components so that we can build our system piece by piece. An object is the fundamental unit (building block) of a system which is used to depict an entity.
- Inheritance - Inheritance is a mechanism by which child classes inherit the properties of their parent classes.
- Abstraction - Mechanism by which implementation details are hidden from user.
- Encapsulation - Binding data together and protecting it from the outer world is referred to as encapsulation.
- Polymorphism - Mechanism by which functions or entities are able to exist in different forms.

UML Diagrams is a rich visualizing model for representing the system architecture and design. These diagrams help us to know the flow of the system. Some of them are:

- 1) Use case diagram
  - 2) Class diagram
  - 3) Sequence diagram
  - 4) Activity diagram
  - 5) Class Diagram
- Use case Diagram:  
A Use Case Diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Interaction

among actors is not shown on the use case diagram.

- Use cases: A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- Actors: An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

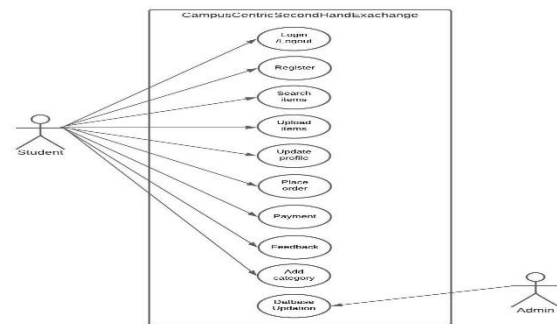
Three relationships among use cases are used often in practice.

- Include: In one form of interaction, a given use case may include another. "Include" is a Directed Relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case. This is useful for extracting truly common behaviors from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "<include>". There are no parameters or return values.
- Extend: In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "<extend>".
- Identified Use Cases: The "user model view" encompasses a problem and solution from the perspective of those individuals whose problem the solution addresses. The view presents the goals and objectives of the problem owners and their requirements of the solution. This view is composed of "use case diagrams". These diagrams describe the functionality provided by a system to external integrators. These diagrams contain actors, use cases, and their relationships.

In our example the preliminaries are,

- Actors: Admin, Student
- Use cases: login, register, view orders, search items, place order, payment,

feedback, add category, upload items, update profile.



**Figure 1:** Use case diagram of Campus Centric Second Hand Exchange

- Class Diagram:  
A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code. It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

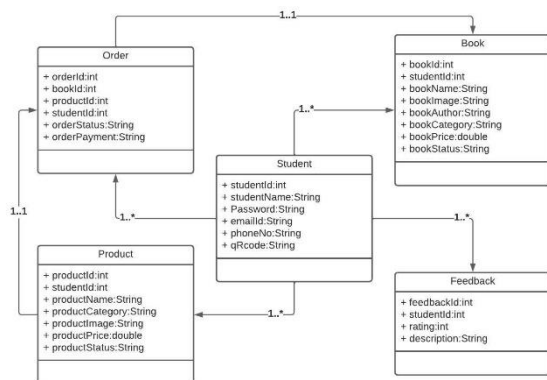
The class diagram is made up of three sections:

- Upper Section: The upper section encompasses the name of the class. A class is a representation of similar objects that shares the same relationships, attributes, operations, and semantics. Some of the following rules that should be taken into account while representing a class are given below:
  - a) Capitalize the initial letter of the class name.
  - b) Place the class name in the center of the upper section.
  - c) A class name must be written in bold format.
  - d) The name of the abstract class should be written in italic format.

- Middle Section: The middle section constitutes the attributes, which describe the quality of the class. The attributes have the following characteristics:
  - a) The attributes are written along with its visibility factors, which are public (+), private (-), protected (#), and package (~).
  - b) The accessibility of an attribute class is illustrated by the visibility factors.
  - c) A meaningful name should be assigned to the attribute, which will explain its usage inside the class.
- Lower Section: The lower section contains methods or operations. The methods are represented in the form of a list, where each method is written in a single line. It demonstrates how a class interacts with data.

In our example the preliminaries are,

- Classes: Student, Book, Product, Order, Feedback
- Attributes: id, name, mobile, password, qrcode



**Figure 2:** Class Diagram of Campus Centric Second Hand Exchange

- Sequence Diagram:  
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the

functionality of the scenario. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

Sequence Diagram Notations:

- Actors – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.

- Lifelines – A lifeline is a named element which depicts an individual participant in a sequence diagram. So, basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

The standard in UML for naming a lifeline follows the following format –

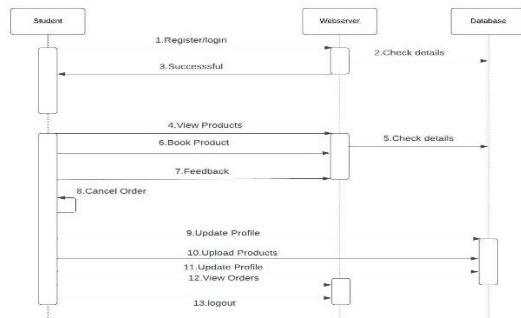
Instance Name: Class Name

- Messages – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram. Messages can be broadly classified into the following categories:

- a) Synchronous messages
- b) Asynchronous Messages
- c) Create message
- d) Delete Message
- e) Self-Message
- f) Reply Message
- g) Found Message
- h) Lost Message

- Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints

attached to a system or a particular process.



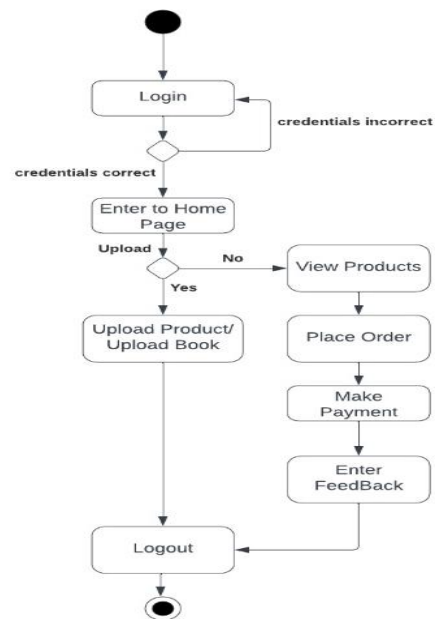
**Figure 3:** Sequence Diagram of Campus Centric Second Hand Exchange

- **Activity Diagram:**  
Activity Diagram is another diagram in UML to describe dynamic aspects of the system. Activity Diagram is basically a flow chart to represent the flow from one activity to another.

Activity Diagram Notations:

- **Initial State** – The starting state before an activity takes place is depicted using the initial state.
- **Action or Activity State** – An activity represents execution of an action on objects or by objects. We represent an activity using a rectangle with rounded corners.
- **Action Flow or Control flows** – Action flows or Control flows are also referred to as paths and edges. They are used to show the transition from one activity state to another.
- **Decision node and Branching** – When we need to make a decision before deciding the flow of control, we use the decision node.
- **Decision node and Branching** – When we need to make a decision before deciding the flow of control, we use the decision node.
- **Guards** – A Guard refers to a statement written next to a decision node on an arrow sometimes within square brackets.
- **Fork** – Fork nodes are used to support concurrent activities.
- **Join** – Join nodes are used to support concurrent activities converging into one. For join notations we have two or more incoming edges and one outgoing edge.

- **Merge or Merge Event** – Scenarios arise when activities which are not being executed concurrently have to be merged. We use the merge notation for such scenarios. We can merge two or more activities into one if the control proceeds onto the next activity irrespective of the path chosen.
- **Swimlanes** – We use Swimlanes for grouping related activities in one column. Swimlanes group related activities into one column or one row. Swimlanes can be vertical and horizontal. Swimlanes are used to add modularity to the activity diagram.
- **Time Event** – We can have a scenario where an event takes some time to complete. We use an hourglass to represent a time event.
- **Final State or End State** – The state which the system reaches when a particular process or activity ends is known as a Final State or End State. We use a filled circle within a circle notation to represent the final state in a state machine diagram. A system or a process can have multiple final states.



**Figure 3:** Activity Diagram of Campus Centric Second Hand Exchange

- **Collaboration Diagram:**  
A Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send receive message. A collaboration diagram is very similar to sequence diagram.

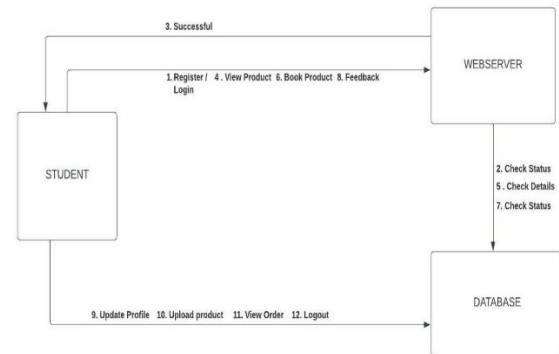
Collaboration diagram shows the objects and their association with other objects. Sequence diagrams and collaboration diagrams shows same information but sequence diagram focus on the temporal aspect and collaboration diagram focus on communication between the objects of system.

In UML diagrams, a collaboration is a type of structured classifier in which roles and attributes co-operate to define the internal structure of a classifier. You use a collaboration when you want to define only the roles and connections that are required to accomplish a specific goal of the collaboration.

### Notations of a Collaboration Diagram

- **Objects:** The representation of an object is done by an object symbol with its name and class underlined, separated by a colon. In the collaboration diagram, objects are utilized in the following ways:
  - The object is represented by specifying their name and class.
  - It is not mandatory for every class to appear.
  - A class may constitute more than one object.
  - In the collaboration diagram, firstly, the object is created, and then its class is specified.
  - To differentiate one object from another object, it is necessary to name them.
- **Actors:** In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.
- **Links:** The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.
- **Messages:** It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labelled

arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.



**Figure 5:** Collaboration Diagram of Campus Centric Second Hand Exchange

- **ER Diagram:**
  - ER model allows you to draw Database Design
  - It is an easy-to-use graphical tool for modeling data
  - It is a GUI representation of the logical structure of a Database
  - It helps you to identifies the entities which exist in a system and the relationships between those entities

### ER Diagrams Symbols and Notations

These are mainly containing three basic symbols which are rectangle, oval and diamond to represent relationships between elements, entities and attributes. There are some sub-elements which are based on main elements in ERD Diagram. ER Diagram is a visual representation of data that describes how data is related to each other using different ERD Symbols and Notations.

Following are the main components and its symbols in ER Diagrams:

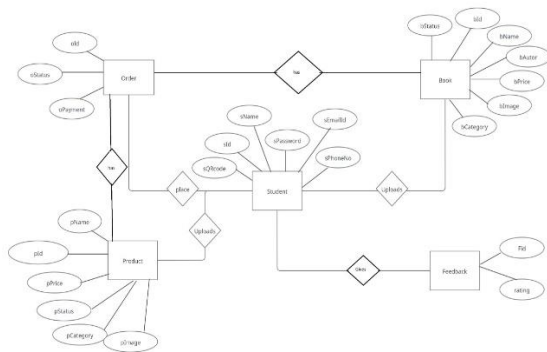
- **Rectangles:** This Entity Relationship Diagram symbol represents entity types
- **Ellipses:** Symbol represent attributes
- **Diamonds:** This symbol represents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined.



- Double Ellipses: Represent multi-valued attributes

This model is based on three basic concepts:

- **Entities:** A real-world thing either living or non-living that is easily recognizable and nonrecognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world. An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.
- **Attributes:** It is a single-valued property of either an entity-type or a relationship-type. For example, a lecture might have attributes: time, date, duration, place, etc. An attribute in ER Diagram examples, is represented by an Ellipse
- **Relationship:** Relationship is nothing but an association among two or more entities. E.g., Tom works in the Chemistry department. Entities take part in relationships. We can often identify relationships with verbs or verb phrases.



**Figure 6:** ER Diagram of Campus Centric Second Hand Exchange

## V. SYSTEM IMPLEMENTATION

### A. Introduction

Nowadays, according to our university conditions, the students usually generate a great deal of spare items such as books, clothes or comedies. Especially for graduating students who are leaving soon. Their quilt, 2 pillow, mattress, washbasin, kettles and so on are inconvenient to move back

home. Just throwing them away is not only a big waste of money but also wastes resources.

Freshmen, the exact people who need these items, could buy these second-hand items. Therefore, an appropriate way to deal with these things is needed. Considering that, this project designs a website to offer service in order to make the trade safe and convenient. The service could also minimize the cost to our environment.

Existing websites do have fundamental customers. There are still some limitations. Since these platforms are open to the whole citizens, we cannot easily find what we need for school and the transportation is not convenient. We are trying to not move heavy items back to home. It makes no difference if they are sold locally or in a remote place? In this case, a trade website which just focuses on a single university is necessary.

Firstly, in the project the research on the environment and conditions of second-hand trading nowadays, was conducted. The positive attitude about the application was obtained after data collection of students thoughts. Secondly, the ideas from the students helped design the interface and functions. The goal of the application would be achieved to help the users deal with daily problems and satisfy them as most as possible.

### B. Project Module

The system consists of three major modules with their sub-modules as follows.

Admin:

- Maintain Database like add, delete, update.
- Monitors all the work.

Student:

- Register: Student can register and obtain credentials.
- Login: Student can login using credentials.
- Forget password: Send an email with OTP and reset password.
- Search for Products.
- Upload Products with image and details
- View and can Update Profile.
- Place and view orders

- View the uploaded products
- Provide rating and feedback

## VI. SYSTEM TESTING

### A. Introduction

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested.

Software Testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for III planned through testing.

#### Testing objectives

These are several rules that can save as testing objectives:

- Testing is a process of executing program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.

#### Test Levels

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or darkness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product.

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

### B. Testing Methods

- **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.
- **Integration Testing:** Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.
- **Functional Testing:** Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.
- **System Testing:** 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.
- **White Box Test:** White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.
- **Black Box Test:** 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.
- **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.
- **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.

- Acceptance Testing: User Acceptance Testing is a critical phase of any project and requires significant participation by the end user.

### C. Test Cases

S. No	Test Cases	Result
1.	Email Id: soumyaparuchuri2000@gmail.com Name: Soumya Password: password Mobile: 912345678 Payment: Offline	Failed Invalid User (Entered mobile number is of less than 10 digits)
2.	Email Id: gchyuigyhbhu Name: Soumya Password: password Mobile: 912345678 Payment: Offline	Failed Invalid User (Entered mail is not valid)
3.	Email Id: soumyaparuchuri2000@gmail.com Name: Soumya Password: password Mobile: 9123456780 Payment: Offline	Passed Valid User

**Table 1:** Test cases for Register Student

S. No	Test Cases	Result
1.	Email Id: soumyaparuchuri2000@gmail.com Password: password	Passed Valid User
2.	Email Id: gchyuigyhbhu Password: password	Failed Invalid User (Entered credentials are not registered)

**Table 2:** Test cases for Login Student

S. No	Test Cases	Result
1.	Name: Wings of fire Author: APJ Abdul Kalam Price:500 Category: Story Image: wingsoffire.jpg	Book Uploaded Successfully

2.	Name Programming Author: APJ Rhema Thereja Price:600 Category: Other Category Name: Computers Image: wingsoffire.jpg	Book Uploaded Successfully
----	---	----------------------------------

**Table 3:** Test Cases for Upload book

S. No	Test Cases	Result
1.	Name: Drafter Price:500 Category: Electronic Device Image: calci.jpg	Product Uploaded Successfully
2.	Name: Chair Price: 600 Category: Daily use Image: chair.jpg	Product Uploaded Successfully

**Table 4:** Test Cases for Upload Product

S. No	Test Cases	Result
1.	Category: Books Keyword: wing	Image: wingsoffire.jpg Name: Wings of Fire Author: APJ Abdul Kalam
2.	Category: Electronic and daily use Keyword: calc	Image: calci.jpg Name: calculator
3.	Category: Books Keyword: electrical	No results found

**Table 5:** Test Cases for Search

S. No	Test Cases	Result
1.	Rating:4 Description: Good website to upload items and this website provides affordable products to purchase.	Feedback uploaded successfully
2.	Rating:5 Description: Good	Feedback uploaded successfully

**Table 6:** Test Cases for Feedback

## VII. CONCLUSION

The development of the Internet greatly changes people's life, and brings a lot of convenience. This system is to develop a second-hand market application system for students, which uses the Internet characteristics. It provides a quick and convenient online trading platform for current students in school, which can let the students spare yard to realize its value, let the students in need access to cheap and satisfied second-hand items. Give full play to its performance, save the costs, and make reasonable use of network resources.

School students now do not have a very good handling for second-hand resources, which not only wastes the resources, also can't save economic cost. Based on these characteristics,

this website designs campus second-hand market application system, systematically expounds the implement and main functions of campus second-hand market application system.

The website designed and implemented commodity information browsing, commodity classification inquiries, online communication and background management system, and other functions. It provides a quick and convenient online trading platform mainly for student's second-hand goods, let the students spare yard realize its value, and let the students in need access to cheap and satisfied second-hand items. It is both environmental and economic, and has high practical value.

List of objectives that are implemented are:

- Student can upload the product details to sell.
- Student can search the product by its keyword.
- Student can place order for a particular product.
- Students can share their feedback to the website.

The aim of developing this project is to provide a platform for students to effortless trading of old products. It not only saves students money but also helps to safeguard the environment.

## VIII. REFERENCES

- 1) "Why Do College Books Cost So Much?", thoughtco, 2017. [Online]. Available: <https://www.thoughtco.com/why-are-textbooks-so-expensive-78849>. [Accessed: 14- Mar- 2018].
- 2) "Second Hand Trading Website for Student", Google, 2018. [Online]. Available: [https://www.google.com.hk/search?safe=strict&client=aff-cs-360se&hs=OWF&ei=BTyiWtjRCMuAjwOEirzAAw&q=second+hand+trading+website+for+students&oq=second+hand+trading+websit e+for+stu&gs\\_l=psy-ab.3.0.33i160k113.292603.294576.0.296320.8.8.0.0.0.209.792.6j1j1.8.0.... 0... 1.1.64.psy-ab..0.3.283...0.MmE3Tm9ZiU4](https://www.google.com.hk/search?safe=strict&client=aff-cs-360se&hs=OWF&ei=BTyiWtjRCMuAjwOEirzAAw&q=second+hand+trading+website+for+students&oq=second+hand+trading+websit e+for+stu&gs_l=psy-ab.3.0.33i160k113.292603.294576.0.296320.8.8.0.0.0.209.792.6j1j1.8.0.... 0... 1.1.64.psy-ab..0.3.283...0.MmE3Tm9ZiU4). [Accessed: 19- Mar- 2018].
- 3) "Sustainable School Shop - Second hand Textbooks, Uniforms etc.", Sustainableschoolshop.com.au, 2018. [Online]. Available: <https://www.sustainableschoolshop.com.au/>. [Accessed: 17- Feb- 2018].
- 4) "Angular Documentation for project development" [Online]. Available: <https://www.typescriptlang.org/docs/handbook/angular.html>
- 5) "Eclipse documentation for backend" [Online]. Available: <https://www.eclipse.org/m2e/documentation/m2e-documentation.html>
- 6) "Hibernate documentation for database access" [Online]. Available: [https://www.tutorialspoint.com/hibernate/hibernate\\_quick\\_guide.htm](https://www.tutorialspoint.com/hibernate/hibernate_quick_guide.htm)
- 7) "Java Textbook" [Online]. Available: <https://gfgc.kar.nic.in/sirmv-science/GenericDocHandler/138-a2973dc6-c024-4d81-be6d-5c3344f232ce.pdf>