INTRODUCTION TO PROJECT

Access to adequate space, at home or office, is always an issue. Unused items tend to lie around and kill productivity. The traditional models of warehousing and other transit service providers leave little options and flexibility for business owners and individuals.

Our storage facilities are designed to offer economical and easily accessible storage space for household and business customers. We feature affordable, convenient and secure facilities to care about your belonging as much as you do.

Our Storage system offers flexibility and convenience when you pick a storage option. Your items stay secure, and you can access them whenever you like. There are restrictions on who can get access to your storage space so that you can stay worry-free.

The system provides you login facility that provides you details about requests. But if user is new, he can first signup and then look for the storage options.

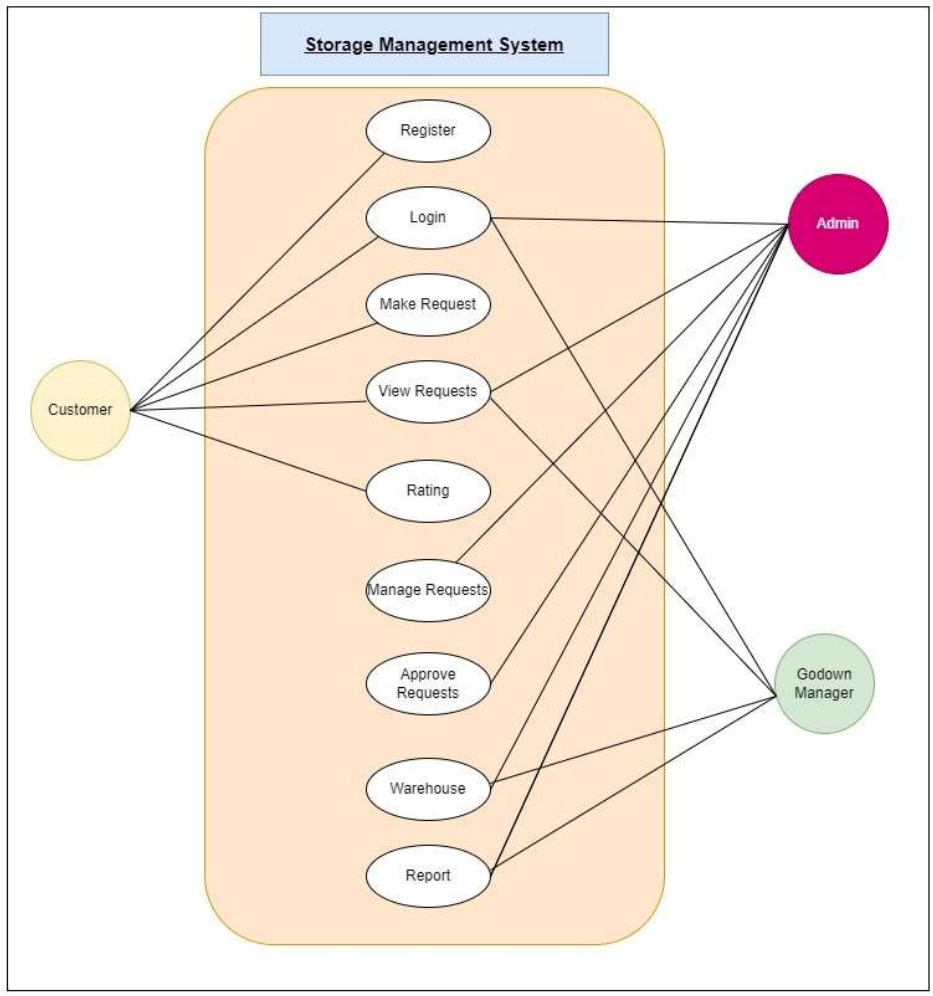
The admin can view all the customers, their requests, the managers, the warehouses, and the capacity. Whenever a customer makes a request, it goes through two step verification that is through manager and then admin.

Our System is an online software application which fulfills the requirement of a typical Stock Analysis in various warehouses. It provides the interface to users in a graphical way to manage the daily transactions as well as historical data. Also provides the management reports like monthly inwards, monthly deliveries and monthly returns.

This application maintains the centralized database so that any changes done at a location reflects immediately. This is an online tool so more than one user can login into system and use the tool simultaneously. The aim of this application is to reduce the manual effort needed to manage transactions and historical data used in various warehouses. Also, this application provides an interface to users to view the details like the daily Stock Statements of all warehouses.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS



The Storage Management system will have a functionality for three types of Users:

1) Admin

2) Customers

3) Warehouse Manager

2.1.1 Admin Account

The admin can manage and view all the transactions on a regular basis. Add or remove customers and warehouse managers. The admin will have access to real-time detailed reports on the incoming and outgoing of the inventory, also accept incoming requests for storage from customers.

2.1.2 User Account

The customer, who will henceforth be called the 'user', will be presented with 2 choices by our system, as the first step in the interaction between them. A user can choose one of these and his choice would be governed by whether he is a new or a registered user. The terms 'registered user' and 'new' are described below.

A user who has used our application earlier would be given a user id and a password. This 'personal information' would be henceforth referred to as 'profile'. A user with a profile in database shall be called a 'registered user'. A registered user will be able to check the availability of storage as well as book storage by logging into the system.

A new user, on the other hand would have to register himself with the system by providing personal information.

'Availability of storage' refers to viewing the capacity, the price of rent and discount offers. The system shall present the user with an option to exit (logout) from the system at any time during the following processes.

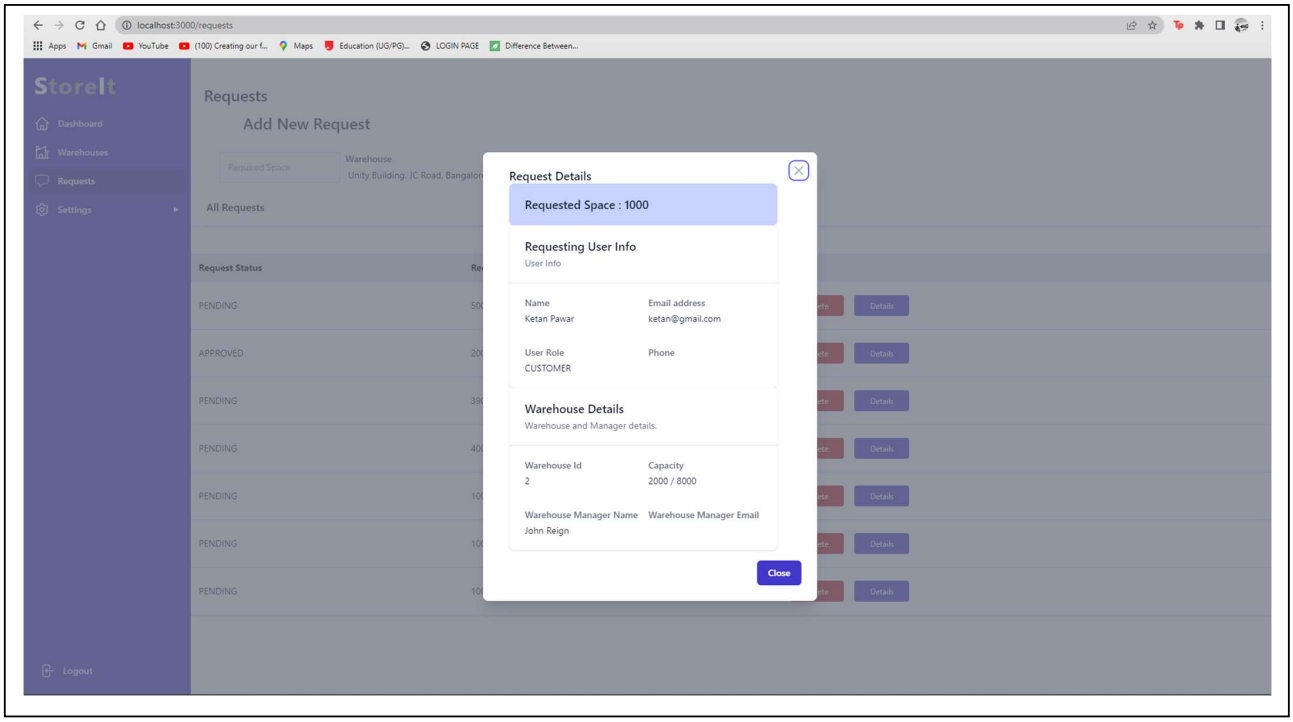
2.1.3 Registration and creation of user profile

The system shall require a user to register, in order to carry out any transactions with it . It will ask the user for the following information – an email, a password, phone number, username.

2.1.4Availability of storage

Here we provided Search facility for any user to see available storage in warehouses. This will provide user an option for searching and comparing their options in all warehouses.

After logging in the system shall request the user to enter the following details : warehouse city . “City’ is a generic term and refers to a city or town as the case may be.



2.2 NON FUNCTIONAL REQUIREMENTS

2.2.1 Interface

Go to Appendix B for user interfaces

2.2.2 Performance

- **Number of Concurrent Users:**

ARS shall be able to handle at least 1000 transactions/inquiries per second

2.2.3 Other Requirements:

- **Hardware Interfaces**

The SPMS is expected to function on Intel PIII 900 MHz Processor equivalent or above, 128 MB RAM, 20 GB HDD.

- **Software Interfaces**

The SPMS shall work on MS Windows operating systems family (MS Windows 98, MS Windows NT Workstation, MS Windows 2000, MS Windows XP). It configures to work with Oracle database. This System works on Apache Tomcat server. It uses browser IE 5.0 & above. It uses IIS 5.0 server.

3. DESIGN

3.1 Database Design

The following table structures depict the database design.

Table1: User_Info

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
PRI	user_id	int	4	0
0	email	Varchar(25)	25	1
0	mob_no	Varchar(10)	10	1
0	password	Varchar(20)	20	1
0	user_name	Varchar(20)	20	1
0	user_role	Varchar(30)	30	1
0	loc_id	int	4	1

Table2: Categories

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Pri	category_id	int	4	0
0	category_name	varchar(30)	30	0
0	description	varchar(100)	100	0

Table3: godown_managers

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Pri	mgr_id	int	4	1
3	mgr_email	varchar(50)	50	0
0	mgr_name	varchar(20)	20	1
0	mgr_pass	varchar(50)	50	1

Table4:Locations

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Pri	loc_id	int	4	1
0	city	varchar(20)	20	0
0	line1	varchar(100)	100	0
0	line1	varchar(100)	100	0
0	pincode	int	4	0
0	state	varchar(20)	20	1
0	address_type	Varchar(20)	20	1

Table5:Products

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Pri	product_id	int	4	1
0	volume	bigint		0
0	product_name	varchar(20)	20	0
0	category_id	int	4	0

Table6:Ratings

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Pri	rating_id	int	4	0
0	rating	int	4	0
0	user_id	Int	4	0

Table7:Request

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Pri	rating_id	int	4	0
0	actual_service_date	date		0
0	**genereation_date**	date		0
0	**request_status**	varchar(20)	20	0
0	**request_type**	varchar(255)	255	0
0	**reqd_space**	Bigint	8	0

0	**customer_id**	int	4	0
0	**warehouse_id**	int	4	0

Table8: warehouses;

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Pri	warehouse_id	int	4	0
0	available_capacity	bigint	8	0
0	total_capacity	bigint	8	0
0	mgr_id	int	4	0
0	loc_id	int	4	0

Table9: request_item ;

Key Type/ Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
Pri	item_id	int	4	0
0	quantity	int	4	0
0	req_id	int	4	0
0	prod_id	int	4	0

E-R Diagram and Class Diagram:

APPENDIX A:Entity Relationship Diagram
APPENDIX B:Class Diagram
APPENDIX C:Flow Diagram

4. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	User,Warehouse, Request.	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	addCategory, deleteRequest	Methods should use **verbs** or verb phrases.
Parameter	Camel	userId	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	IUserService	Do not use the '_' sign
Property	Pascal	ForeColor, BackColor	Use a noun or noun phrase to name properties.
Associated private member variable	_camelCase	_userName, _password	Use underscore camel casing for the private member variables
Exception Class	Pascal with "Exception" suffix	ResourceNotFoundException,	

Comments

- Comment each type, each non-public type member, and each region declaration.
- Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
- Separate comments from comment delimiters (apostrophe) or // with one space.
- Begin the comment text with an uppercase letter.
- End the comment with a period.
- Explain the code; do not repeat it.

5. TEST REPORT

Another group called Linux did the testing and the report of the testing is given hereunder.

GENERAL TESTING:

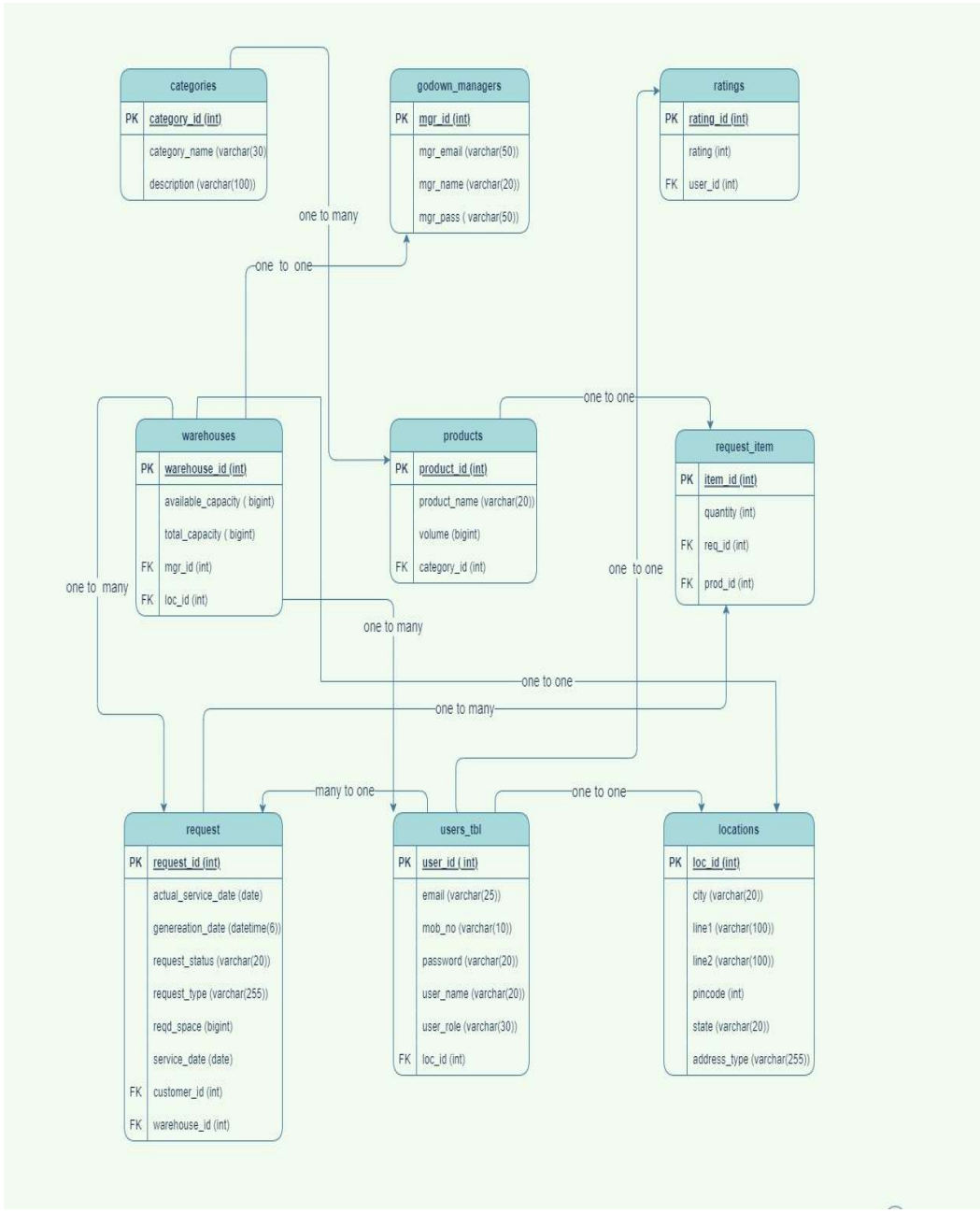
SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	SignUP Page	Redirected to Next page	OK	Nothing
2	Login Page	Pop-up will come	Ok	Please enter username and password again .
4	Homepage	Gives all dashboard	Ok	Nothing
5	List of Customers	All the customers list should be visible	Ok	Nothing
6	Customer Details	Information of customer	Ok	Nothing
7	List of Mangers	All the managers list should be visible	Ok	Nothing
8	Request page	List of all the requests with their status	Ok	Nothing
9	List of Warehouses	List of all the warehouses	Ok	Nothing
10	Make request page	Form to make a request	Ok	Nothing
11	Manage requests	Action to be taken on the request	Ok	Nothing
12	Logout	It will logout user form profile.	Ok	Nothing
	STATIC TESTING			
SR-NO	**Deviation**	**Program**		
1	Commenting not followed	All Web Application		

6. PROJECT MANAGEMENT RELATED STATISTICS

DATE	WORK PERFORMED	*SLC Phase*	Additional Notes
FEB 09,2023	Project Allotment and User Requirements Gathering	Feasibility Study	Our team met the client Mrs. Pooja Bhandare (Faculty, SIIT Pune) to know the requirements.
FEB 12,2023	Initial SRS Document Validation And Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better
FEB 18,2023	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces	Requirement Analysis & Design Phase	Database Design completed
FEB 19,2023	Business Logic Component design Started	Design Phase	----------
FEB 21,2023	Coding Phase Started	Coding Phase	30% of Class Library implemented.
FEB 23,2023	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
FEB 24, 2023	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.
FEB 28, 2023	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	--
MAR 01, 2023	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager

MAR 03, 2023	The Project was Tested by the respective Team Leaders and the Project Manager	Testing Phase (Module Testing)	--
MAR 05, 2023	The Project was Submitted to Other Project Leader of Other Project Group For Testing	Testing Phase (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing
MAR 07, 2023	The Errors Found were Removed	Debugging	The Project was complete for submission
MAR 09, 2023	Final Submission of Project		

Appendix A
Entity Relationship Diagram



Appendix B

Class Diagram