

## CHAPTER 1

### INTRODUCTION

#### 1.1 OVERVIEW:

Shortening URLs is a widely used process nowadays. With the presence of social media channels where the text is needed to adapt to a restricted number of characters, usually 140 characters as per the study, therefore shortening URLs became more crucial. So, to shorten the long URLs, a greater number of URL-shortening resources are being used [1]. When someone needs to shorten their unpleasantly long URL, they go to a URL shortening resource and submit the long URL meant to get shortened [2]. Short URLs are usually no longer than 10 to 15 characters. Instead of using long URLs, users can visit these shortened URLs. This resource handles the connection between the original, long link and the new shortened link, which gives the user a simple link of about 5 to 10 characters instead of the lengthy one. Further, URL shortening services also incorporate link tracking for analytics to track clicks and users. Docketed links permits link masking. Link masking is the consequence of the URL shortening where the long URL changes into a short bone. Masked links are better to track where links appear. This operation can be further enforced to check whether the diverted URL website is secure or not.

#### 1.2 EXISTING SYSTEM:

In the existing system the links are not checked for website availability i.e., if a website is not reachable the user is not notified about it and the link is shorted, which is no use for the user as the link will direct to the error page.

##### Limitations of Existing System:

- Less security
- Invalid links not identified.
- Advanced features are costly.

### **1.3 PROPOSED SYSTEM:**

In proposed method to overcome the limitations of the existing system. There are three sections in this proposed system they are public, private, and custom. The first section is the public section where the link can be shortened and no security is provided as private section, this is created when user doesn't want to restrict the link to specific persons. The link can be used by everyone and the detailed report of which can be seen in the dashboard. The second section is private section, the user can enter the desired link that needs to be shorted and then set a unique password to that link. After doing this, whenever someone wants to use that link, they must enter a valid password that was created during link creation, this private section is provided to the user to give additional security so that one cannot access the link without entering a valid password. This is helpful for the user who wishes to share the link with a specific person who wants to use the link and if any other person gets the link, he/she should not be able to access it. The third section is custom section where the user can create his own link, as in the public or private section the link is randomly shorted by the system by a default character limit of 5, but in custom section the user is given full access to customize the link. In this section the user can create his desired link which is easily remembered by the user rather than the system generated link.

#### **Benefits of Proposed System:**

- More Security
- Invalid links are identified.
- Advanced features are available for free.
- Multi-User Access

### **1.4 PROBLEM STATEMENT AND SCOPE OF THE PROJECT:**

#### **Problem statement:**

To design and develop a fun-size URL and generate a QR code for the shortened URL, this shortened URL will not break while sharing on any social media platforms.

#### **Scope of the project:**

The scope of the project is to shorten the URL and generate QR code for the shortened URL, the generated links will expire after a standard default time.

## **1.5 VISION AND MISSION, OBJECTIVES:**

### **Vision:**

To design and develop a fun-size URL and generate a QR code for Organizations/Institutions.

### **Mission:**

To generate the alias name and QR code for the input URL using the Django web framework.

To fetch the specific URL based on the date of creation using a search algorithm.

To display the URL in a specific order to a user using a sorting algorithm.

### **Objectives:**

1. To minimize the webpage address into something that is easier to remember and track.
2. To generate a QR Code for the URL.
3. To provide a custom URL for the user.
4. To provide a detailed analysis of the URL and QR code with graphical representation.
5. To provide user with end-to-end access of the URL controls.

## CHAPTER 2:

### LITERATURE SERVEY

In article [1], the author offered a URL shortening resource that will take a long URL/Web address and create a shorter one. The Short URL that won't break when we share it across different platforms and make them more manageable using flask framework base62 algorithm.

In article [2], the author discusses the advantages of QR codes over conventional barcodes. QR codes can store information both horizontally and vertically, allowing them to store a much greater amount of information than traditional barcodes. The information encoded in a QR code can be text, a URL, or other data and can be read by mobile device cameras. The popularity of smartphones has expanded the applications of QR codes, which are now used in various industries such as commercial tracking, entertainment and transport ticketing, product marketing, and in-store product labelling.

In article [3], the author offered a composite and vigorous nature of information flow is based on computer server technologies and makes the Web hard to conserve. With this reference this application checks the status code of the website to confirm it is valid or not. There are many status codes available, but status code 404 is the one which will give the information that the site cannot be reached.

In article [4], the author has discussed the fact that link shortening services save space and make manual URL entry cheaper. Although, they create output which is poorly look right on to entry on mobile devices. We identified the issue with the output of a link-shortening service, bit.ly, and presume that longer links that are shortened for input on mobile keyboards would enhance link entry speeds. We conducted a human performance study that confirmed this thesis and applied our method to a selection of different non-word mobile data-entry tasks. This work demonstrates the need for resource design which is suitable for the limitations of the devices public use to consume resources.

In article [5], the author has analyzed the habit of using shortened URLs from creating them to sharing, by using a large-scale dataset of 4.2 B require for 80M URLs created past by Bit.ly. It finds the content URLs that are unknown, making them tough to comprehend. One of the best ways to produce and share online content is using a URL shortening resource, this provides a shortened URL that is redirected to an original URL of content, this application

comprehensively analysis the habit of using shortened URLs from their creations to publishing to sharing.

In article [6], the author introduces a steganographic system that uses fast response (QR) code as a holder to the opponent while transferring its usual message and payload. The QR code which is generated is a normal QR code, making it less open to a holder's attack. The proposed system is space-saving, has a bearable level of noise release, and is subjected to cryptographic attacks. The QR code, which is generated for the link, by scanning it will redirect to the original link.

In article [7], the author has discussed the importance of URL previewing to ensure user confidence when using shortened URLs. It then presents Bit.ly as a freeware and sophisticated URL shortcut resource that not only shorts URLs and provides real-time follow-up, but also offers variety of features, such as a history of shorter URLs, a bookmarklet, a Firefox plugin for previewing, a spreadsheet template, a Gmail gadget, and an API for different applications.

In article [8], the author has discussed the use of short URL services to convey information easily when a long URL with special characters is difficult to share. However, attackers can abuse short URLs to distribute malicious code and launch phishing attacks. The proposed method suggests writing destination information when generating short URLs to permit users to inspect if it is a web document or file. URL shortening resource providers keep track of the risk of target URLs and decide to provide service or not. They assess the risk of web pages and may block short URLs to prevent attacks like "drive-by download."

In article [9], the author has discussed the Status codes are three-digit numerical codes returned by web servers to web browsers to indicate the status of an HTTP transaction. They are categorized into five groups based on the first digit of the code, with the second and third digits providing more specific information. Some status codes include 200 OK, 301 Permanent Redirect and 404 Not Found. These codes are crucial for troubleshooting and debugging web applications, providing information on errors and diagnosing problems with a website's performance. Developers can optimize their websites and enhance user experiences by understanding the different types of status codes.

In article [10], the author has discussed that a URL redirector service is a web-based tool that takes a long URL and converts it into a shorter, more manageable link. When a user clicks the shortcut URL, they automatically move to the original long URL. The process of using a URL-redirector service is straight forward. First, the user enters the long URL they want to shorten

into the service's interface. The redirector service then assigns a unique identifier to the URL and stores it in its database. The service then returns the shortened URL to the user, which they can share or post online. The redirector resource uses the unique identifier assigned to the shortened URL to locate the corresponding lengthy URL in its database and then forwards the user to that URL. URL-redirector services are commonly used to create shorter, more manageable links that are easier to share on social media and in other online contexts. Additionally, they allow users to track clicks and other metrics associated with the URLs they have shortened. This information can be used to measure the effectiveness of marketing campaigns, among other things.

## CHAPTER 3:

### REQUIREMENTS

#### 3.1 FUNCTIONAL REQUIREMENTS:

- Given a URL, our service should generate a shorter and unique alias of it. This is called a short link. This link should be short enough to be easily copied and pasted into applications.
- When users access a short link, our service should redirect them to the original link.
- Users should optionally be able to pick a custom short link for their URL.
- Links will expire after a standard default timespan.

#### 3.2 NON-FUNCTIONAL REQUIREMENTS:

- **Performance:** The Application is built in such a way by taking all the necessary precautions and this as gone through rigorous testing in order to ensure the user a swift feel.
- **Availability:** It will be available only when there is an internet connection because it is an online application.
- **Usability:** The GUI is made so responsive which would make the user to easily work with the software.
- **Portability:** Users can easily create and share shortened links across multiple platforms without any compatibility issues.
- **Maintainability:** The Software is built so simple such that any further updates if required can be implemented easily.

### **3.3 HARDWARE AND SOFTWARE REQUIREMENTS:**

#### **HARDWARE REQUIREMENTS:**

- Processor: I3/Intel Processor
- RAM: 8GB (min)
- Hard Disk: 128 GB

#### **SOFTWARE REQUIREMENTS:**

- Operating System: 64-bit Windows 10.
- Server-side Script: Python 3.9+.
- IDE: Visual Studio Code.
- Web design: HTML, CSS, JavaScript.
- Frameworks: Django web framework.
- Database: SQLite



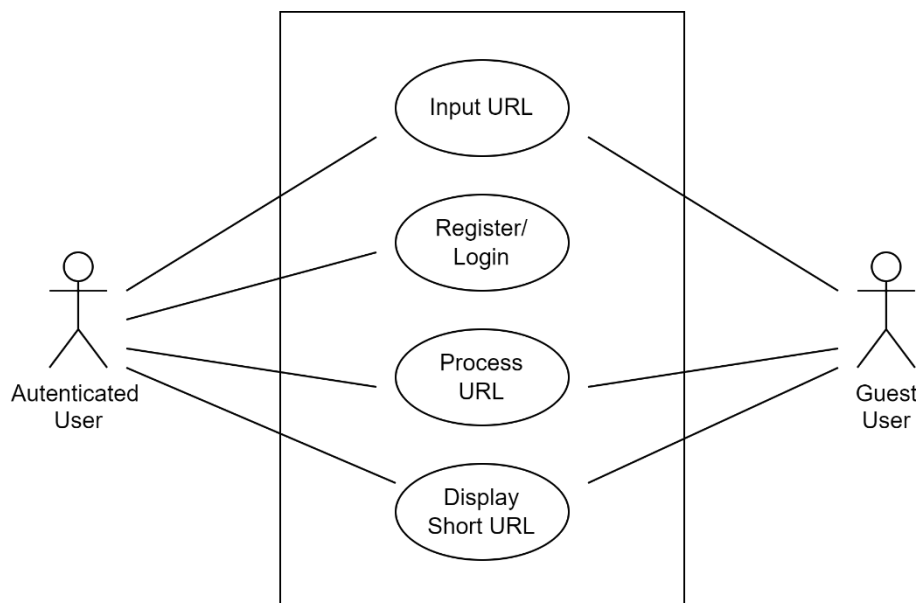
## CHAPTER 4:

### SYSTEM DESIGN

#### 4.1 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.
- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

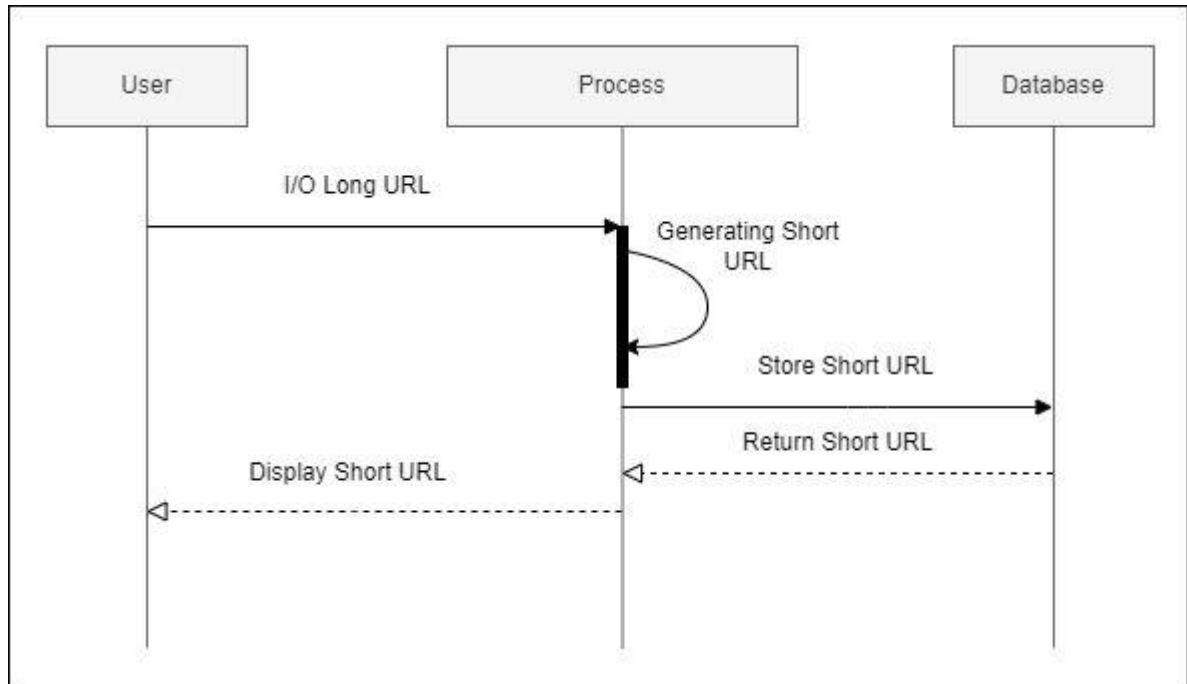
The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



*Fig 4.1.1 use case diagram for the system*

## 4.2 SEQUENCE DIAGRAM:

- A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.
- It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



*Fig 4.2.1 sequence diagram for input URL.*

Below diagram gives the sequential representation of Register/Login of the system.

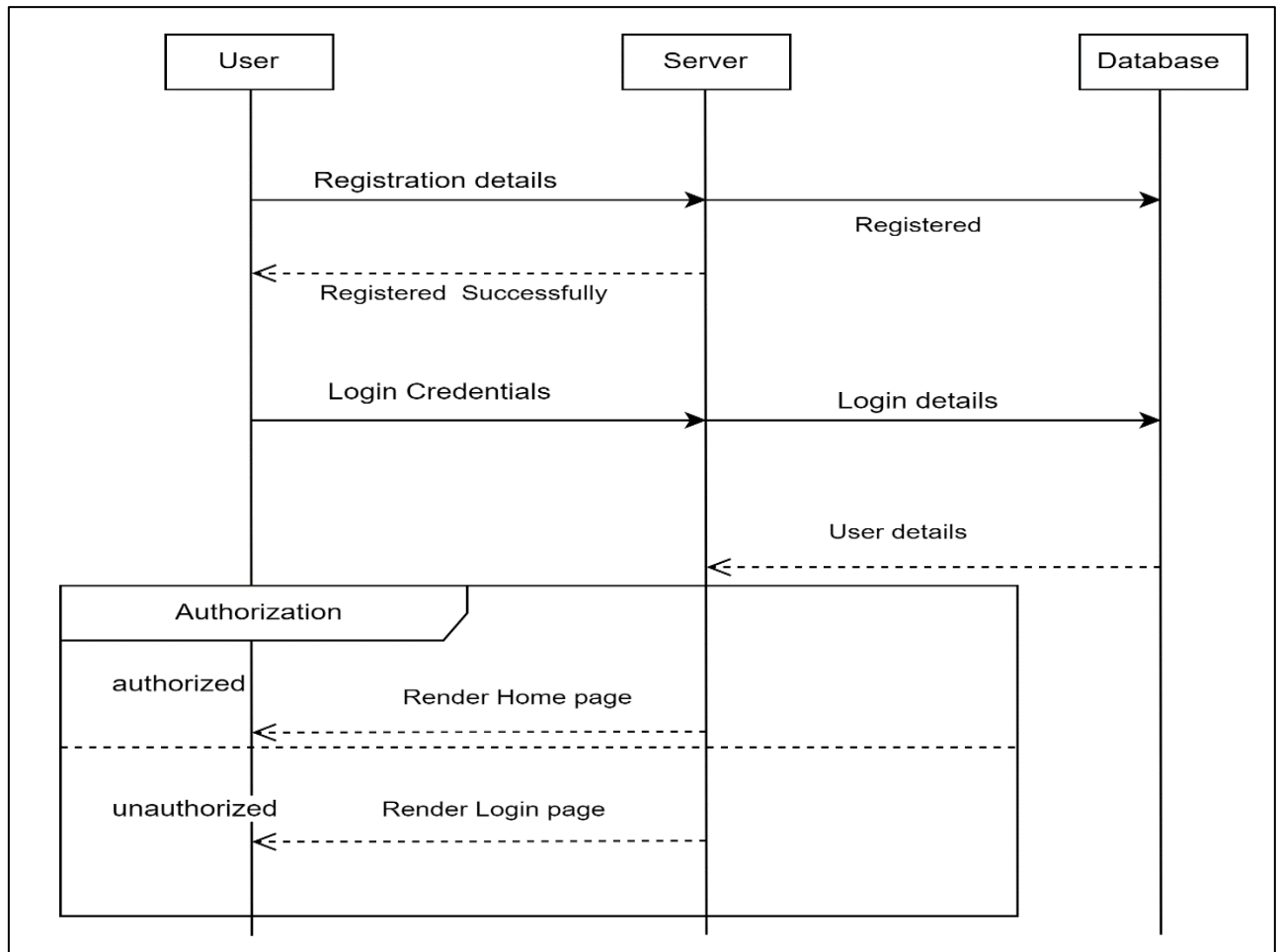


Fig 4.2.2 sequence diagram for login/register.

Below diagram gives the sequential representation of detailed Processing and validating of the status codes of the URL to be shortened.

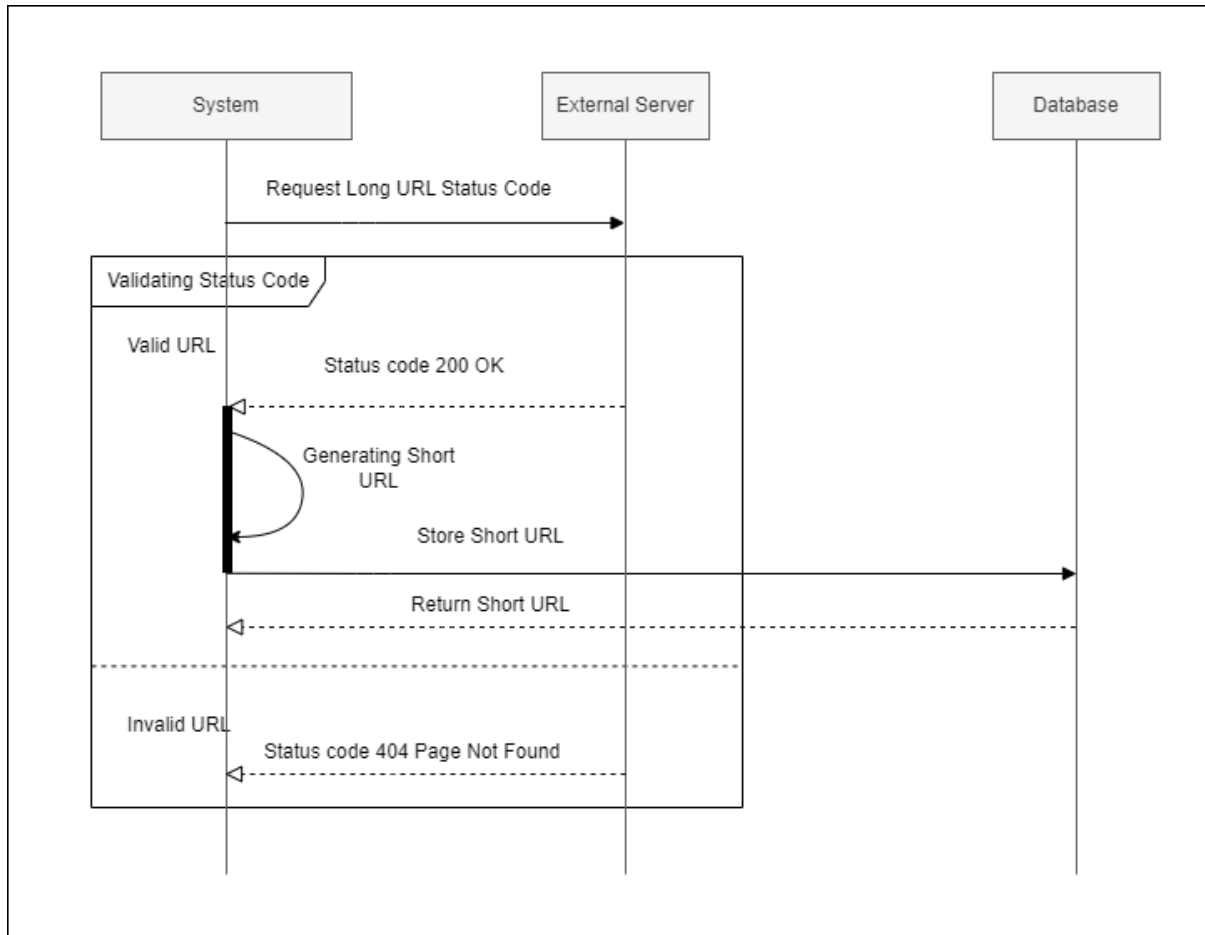
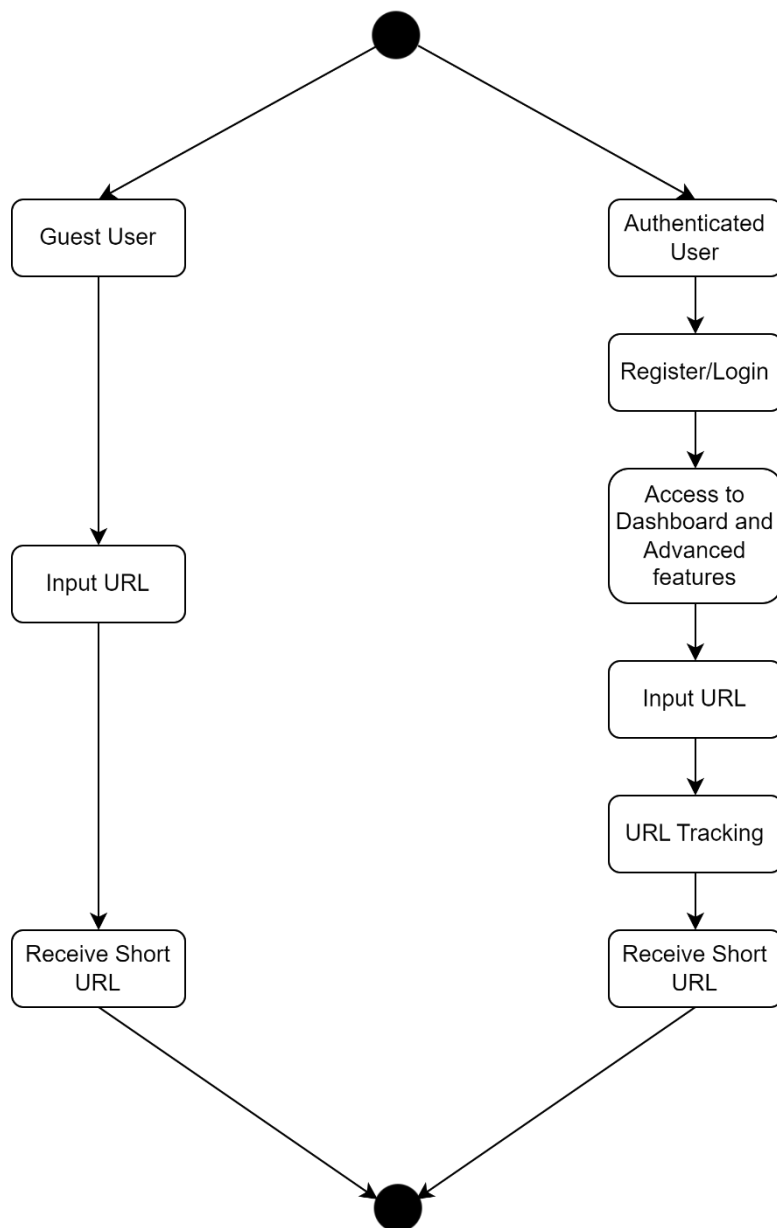


Fig 4.2.3 sequence diagram for processing the URL at server.

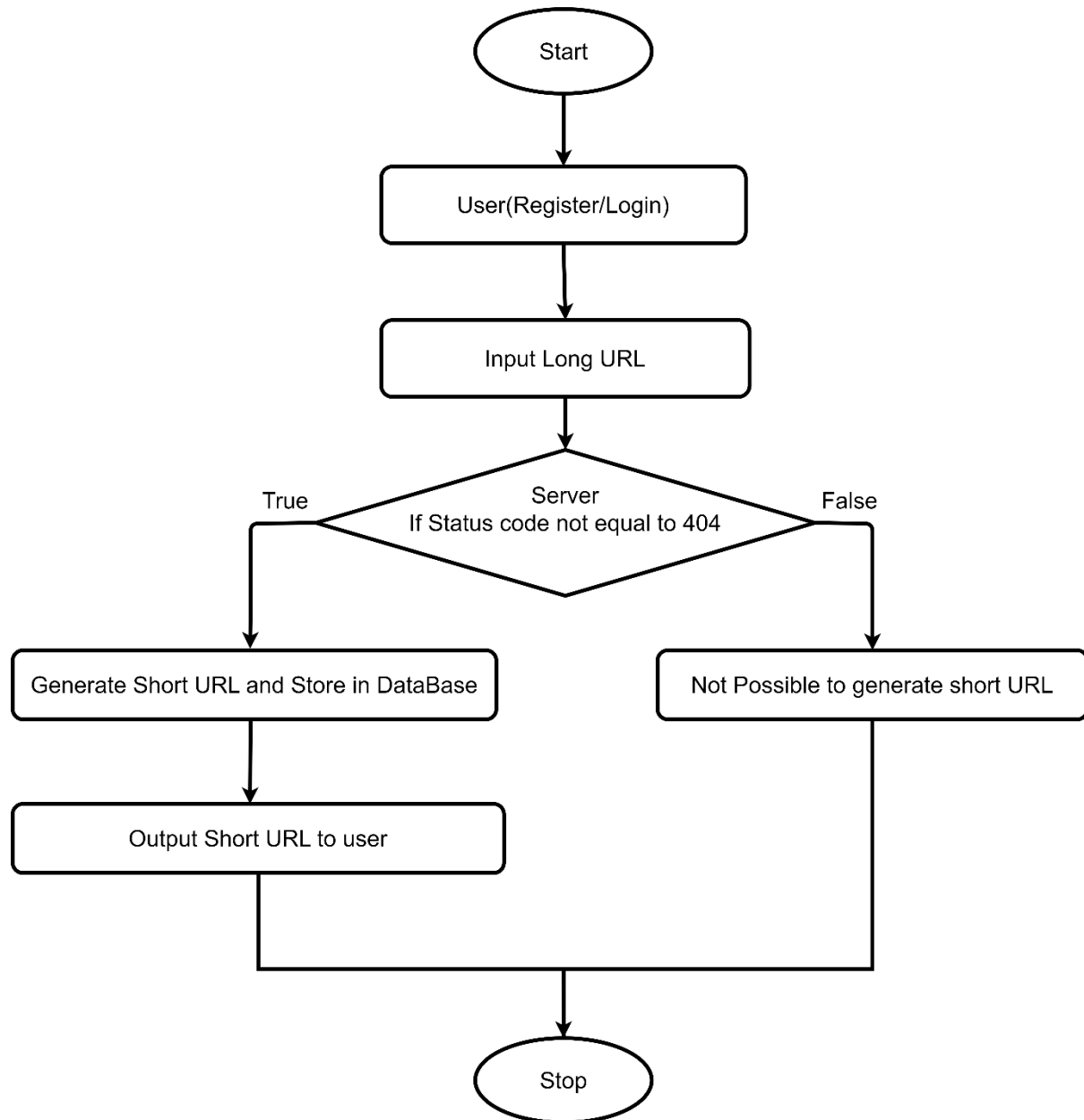
### 4.3 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



*Fig 4.3.1 Activity Diagram*

## 4.4 FLOW CHART



*Fig 4.4.1 Flow chart*

The proposed methodology for shortening the URL is carried out in the steps shown below.

### Step 1: User (Register/Login)

In this the user is allowed to register to create a new account or login to an existing account. After login the user is allowed to input the long URL that need to short.

## Step 2: Check Status Code

In this step the long URL is sent to server which then checks the status code of the long URL. Here a python library urllib takes the URL and checks the status code of that link and gives the output. If the status code of Long URL is 404 then the Server does not convert the long URL into shorter one, it just gives the output as Error:404 page not found for the provided Input URL. If the status code of Long URL is other than 404 then the server converts the long URL into shorter one and stores the original and shorted URL in the database. It also checks for Secure socket layer (SSL) certificate if the URL is SSL certified then only it will short the URL. While checking for status code if the provided URL takes much time to fetch the website, then it will return as Failed page even if it is reachable. Output the Short URL to the user. The output short URL is ready to use.

## 4.5 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

Level 0:

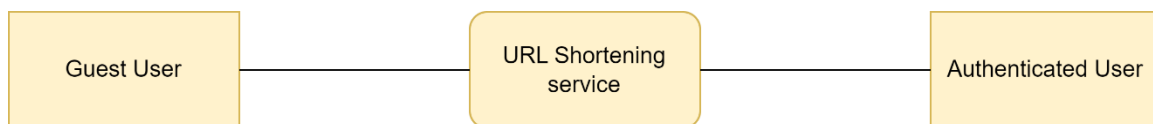


Fig 4.5.1 Level 0 Dataflow Diagram

Level 1:

It indicates how the system is divided into subsection, each of which deals with one or more of the data to or from an external agent, each of which together provide all of the functionality of the system as a whole, it also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



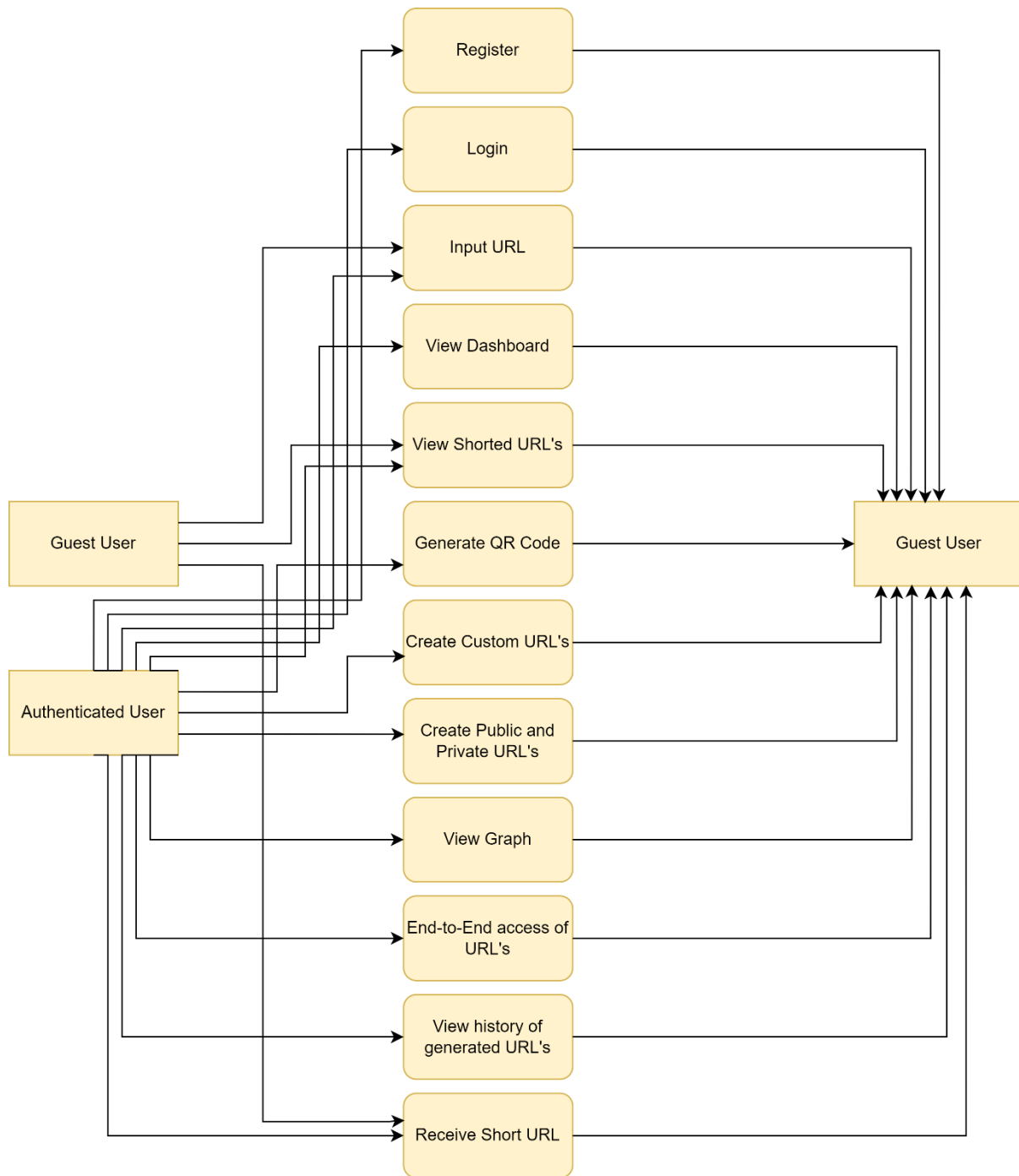


Fig 4.5.2 Level 1 Dataflow Diagram

## 4.6 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 the classes. It explains which class contains information.

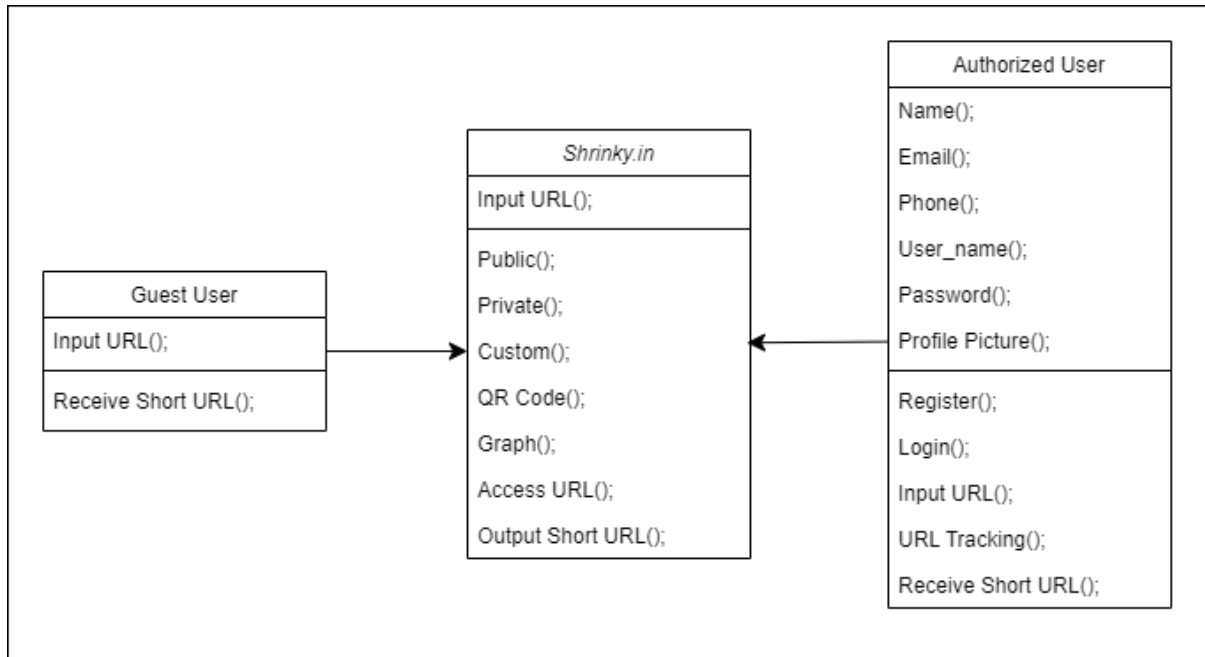


Fig 4.6.1: Class Diagram

## 4.7 ER DIAGRAM:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

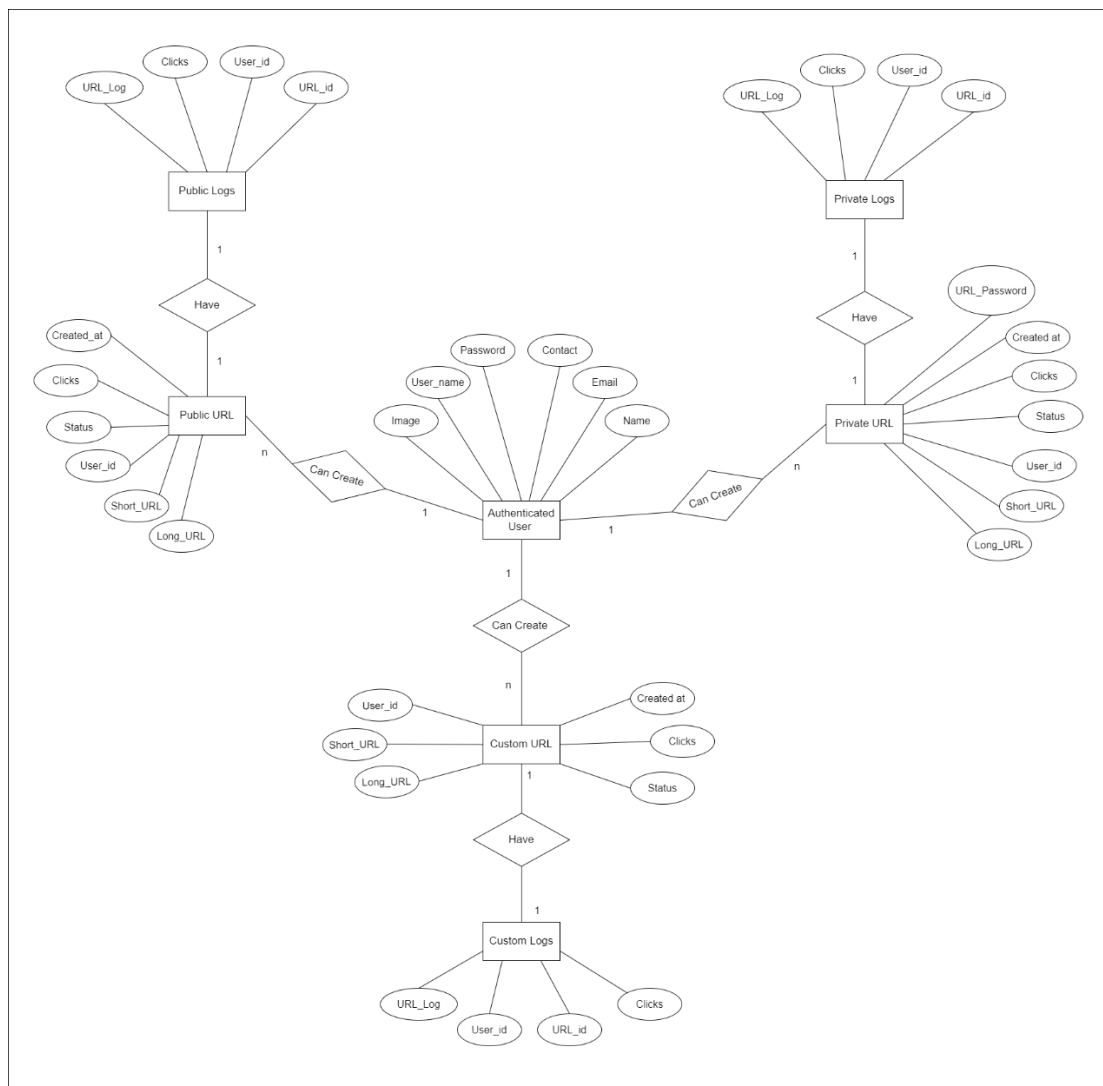


Fig 4.7.1: ER Diagram

## CHAPTER 5:

### IMPLEMENTATION

#### 5.1. SOFTWARE REQUIREMENTS

Following are the Software requirements used.

1. JavaScript
2. Python
3. Django Web Framework
4. SQLite

##### 5.1.1 JavaScript

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called live Script and renamed as JavaScript to indicate its relationship with java. JavaScript supports the development of both client and server components of Web-based applications on the client side, it can be used to write programs that are executed by Web browsers within the context of Web page. On the server side it can be used to write web server programs that can process information submitted by Web browser and then update the browsers display accordingly. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags.

##### 5.1.2 Python

Python is a high-level, interpreted programming language that was first released in 1991 by its creator, Guido van Rossum. It is a popular language used for general-purpose programming, web development, data analysis, artificial intelligence, and more.

One of the reasons Python has become so popular is due to its simplicity and readability. The language is easy to learn, and its syntax is concise and intuitive. Python code can be written quickly and with fewer lines of code than many other programming languages.

Python also has a large and active community of developers who have contributed to the development of numerous libraries and frameworks, making it a versatile language that can be used for a wide range of applications. Some popular libraries include NumPy for numerical computations, Pandas for data manipulation, and Django for web development.

Python's strengths also include its cross-platform support, meaning that code written on one platform can be easily run on another. It also has a vast ecosystem of third-party packages and libraries, which makes it an ideal choice for scientific computing, data analysis, machine learning, and more.

### **5.1.3 Django Web Framework**

Django is a high-level web framework that enables developers to build web applications quickly and efficiently. It follows the model-view-controller (MVC) architectural pattern and is written in Python, a popular and user-friendly programming language.

With Django, developers can focus on the core logic of their applications while taking advantage of the framework's built-in features and conventions. It provides a solid foundation for handling various web development tasks, such as database management, URL routing, form handling, authentication, and more.

### **5.1.4 SQLite**

SQLite is a software library that provides a lightweight, serverless, and self-contained relational database management system (RDBMS). It is widely used as a database engine for small-scale applications, embedded systems, mobile devices, and as a local data storage solution.

Unlike traditional RDBMS systems, SQLite does not require a separate database server process to be running. It operates directly on the client side, allowing applications to interact with the database without the need for network communication or a separate database server installation.

SQLite is easy to set up and use since it requires minimal configuration. It stores the entire database in a single file, typically with a .db or .sqlite extension, making it portable and easy to manage. SQLite supports the essential features of a relational database system, including tables, rows, columns, indexes, and constraints. It allows you to define tables with various data types, create relationships between tables using foreign keys, and enforce data integrity through constraints. SQLite follows the SQL (Structured Query Language) standard and supports a wide range of SQL syntax and commands. It allows you to perform common database operations such as querying data, inserting, updating, and deleting records using SQL statements.

## CHAPTER 6

### MODULES

- **Login window:** This module is the start of the application. It consists of two fields Username and Password by which the application validates verified users who can access the application's benefits. If the user enters a wrong username or password, then the application cannot be used by the user. After logging in the user can use our application. Hence this module plays an important role in the application.
- **Dashboard:** This is the window where the user can start working with the features of the application. The following are the main features:
  - **Generate QR code:** In this module the user can generate a QR code for the shortened URL which, when scanned, redirects to the original URL.
  - **End-to-End access:** In this module the user has end-to-end access of the URL. User can activate or deactivate a particular URL of his choice.
  - **Custom URL's:** In this module the user can create a custom URL meaning user can give any name to the URL of his choice which when used redirects to the original URL.
  - **Graph:** In this module the user can view the total number of URL's generated till date, total number of clicks on those URLs till date and total number of URL's used on that day.
  - **Details of the URL:** In this module the user can view all the links that are shorted till date corresponding with the original links and the status whether they are active or inactive. The user can search for a specific link by using the search bar given.

## CHAPTER 7:

### SYSTEM TESTING

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding a yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is vital to the success of the system.

In the code testing the logic of the developed system is tested. For this every module of the program is executed to find an error. To perform specification test, the examination of the specifications stating what the program should do and how it should perform under various conditions.

Unit testing focuses first on the modules in the proposed system to locate errors. This enables us to detect errors in the coding and logic that are contained within the module alone. Those resulting from the interaction between the modules are initially avoided. In unit testing step each must be tested separately.

Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implementation system. The software must be executed several times in order to find the errors in the different modules of the system.

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors. The validation phase reveals the failures and the bugs in the developed system. It will come to know about the practical difficulties the system faces when operated in the true environment.

Testing may be done at different levels.

- Unit level
- Module level
- Integration and system level

## 7.1 UNIT TESTING:

TEST ID	USER	TESTCASE NAME	INPUT	EXPECTED OUTPUT
T1	End User	Login	Valid Credential	Successful Login
			Invalid Credential	Unsuccessful Login
T2	End User	Register	Username (Unique)	Successfully Registration
			Username (Already Exists)	Invalid (Try different username)
T3	End User	URL Validation	Valid URL	Generate Short URL
			Invalid URL	Prompt back as Invalid URL
T4	Guest User	Validating URL Expiry	If URL not expired	Redirect to original URL
			If URL expired	Page 404
T5	Authorized User	Private URL (Authenticate with password)	Valid Password	Redirect to original URL
			Invalid Password	Page 404

Fig 7.1.1 Unit Testing

## 7.2 MODULE LEVEL TESTING

Module testing is done at each module using test cases as prepared above. Module level testing examines the output of every module involved. Modules are designed during the time of design.

## 7.3 INTEGRATION AND SYSTEM TESTING

Integration testing is used to verify the combining of the software modules. Integration testing addresses the issues associated with the dual problems of verification and program construction. System testing is used to verify whether the developed system meets the requirements.



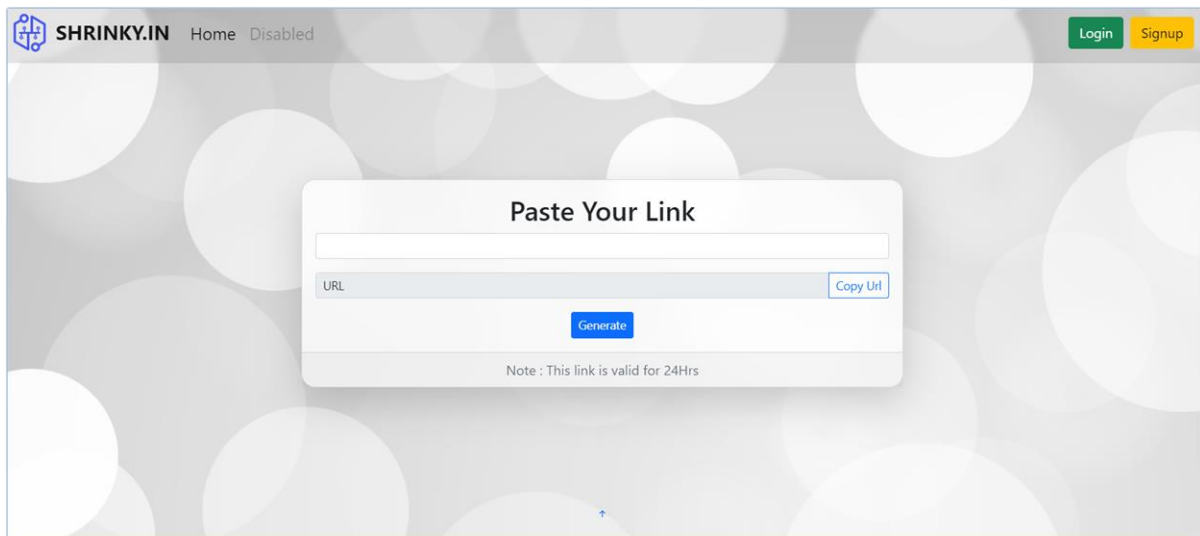
## CHAPTER 8:

### RESULTS AND DISCUSSION

Following the necessary requirement gathering, the proposed system is demonstrated by considering two users like the guest user and authenticated user. The execution is based on those who want to use only the most basic features and those who want more information about the link's functionality.

#### 8.1 Guest Users:

The Fig:4.1 depicts the visibility for guest user, the user can short the URL by pasting the link in the specified field and should click on the generate button, if the link is valid then the short link is generated in the field, user can copy the link by clicking on copy URL button, the link will be copied to the clipboard. This link is now ready to use and is valid for 24hrs.

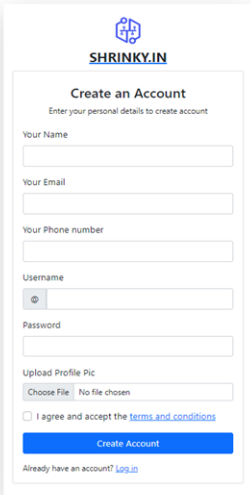


*Fig 8.1.1 Guest user page*

## 8.2 Authenticated Users:

### Register page:

The Fig:4.2 shows the visibility for authenticated user, the authenticated users can Register to create an account by filling up the below form.

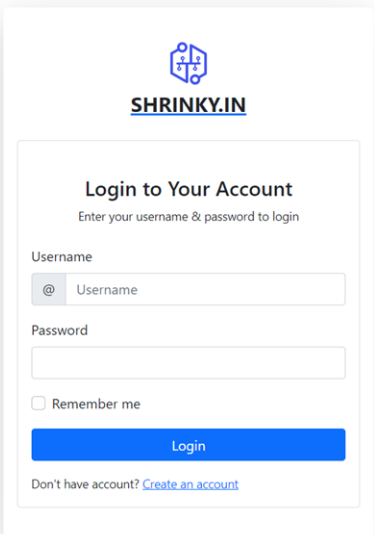


The screenshot displays the 'Create an Account' registration form for SHRINKY.IN. The form is titled 'Create an Account' with a subtitle 'Enter your personal details to create account'. It includes input fields for 'Your Name', 'Your Email', 'Your Phone number', 'Username', and 'Password'. There is a section for 'Upload Profile Pic' with 'Choose File' and 'No file chosen' options. A checkbox for 'I agree and accept the [terms and conditions](#)' is present. A blue 'Create Account' button is at the bottom. A link for 'Already have an account? [Log In](#)' is at the very bottom.

*Fig 8.2.1 Register page*

### Login page:

The Fig:4.3 shows the visibility for authenticated user, the authenticated users can login by providing the credentials in the login form.



The screenshot displays the 'Login to Your Account' form for SHRINKY.IN. The form is titled 'Login to Your Account' with a subtitle 'Enter your username & password to login'. It includes input fields for 'Username' and 'Password'. There is a 'Remember me' checkbox. A blue 'Login' button is at the bottom. A link for 'Don't have account? [Create an account](#)' is at the very bottom.

*Fig 8.2.2 Login page*

## Dashboard:

The Fig:4.4 shows the visibility for authenticated user. this is the window where the user can start working with the features of the application.

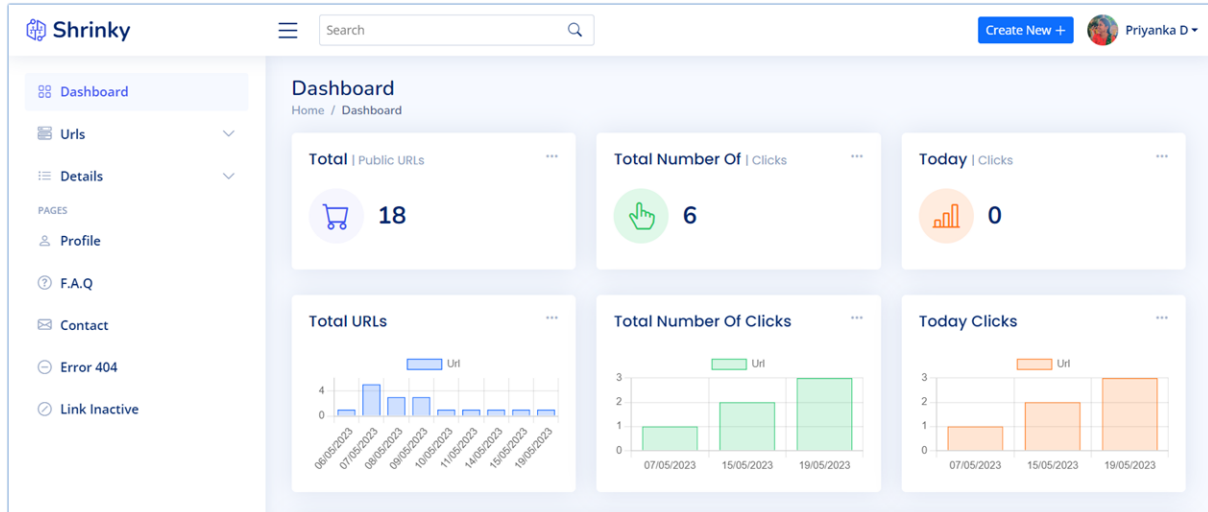


Fig 8.2.3 Dashboard

## Public Section:

The Fig:4.5 shows the visibility for authenticated user, in this section, the link can be shortened, and no security is provided for the link.

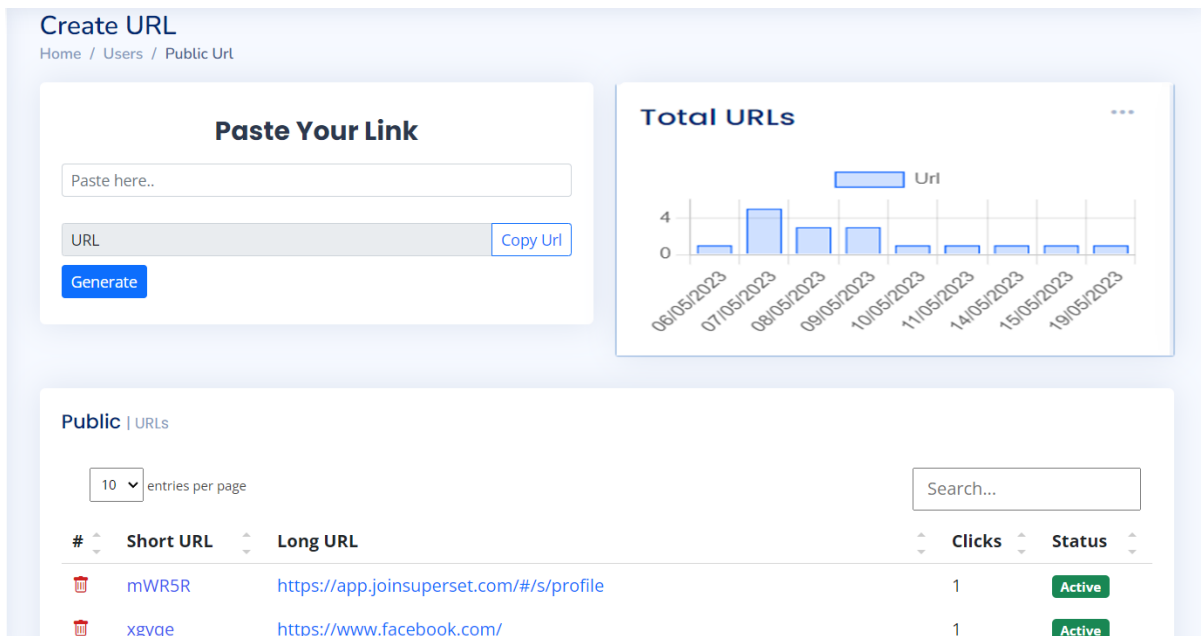


Fig 8.2.4 Public section page

## Private Section:

The Fig:4.6 shows the visibility for authenticated user, in this section, the user can enter the desired link that needs to be shorted and then set a unique password to that link.

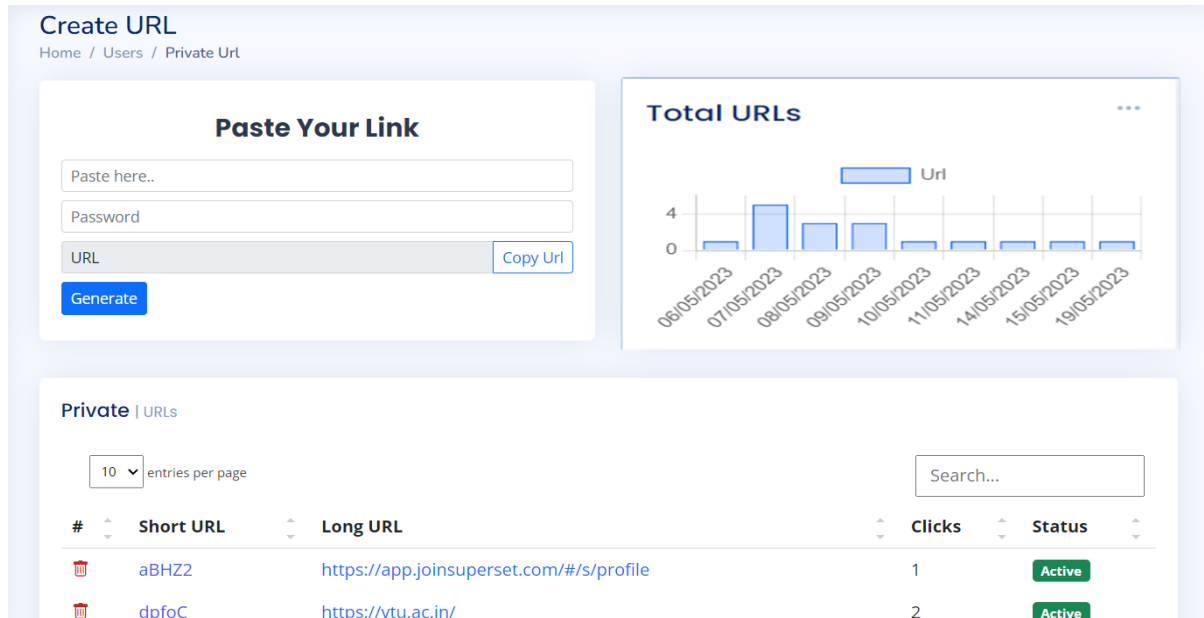


Fig 8.2.5 Private section page

## Custom Section:

The Fig:4.7 shows the visibility for authenticated user, in this section, the user can customize his own link, which is easily remembered by the user rather than the system generated link.

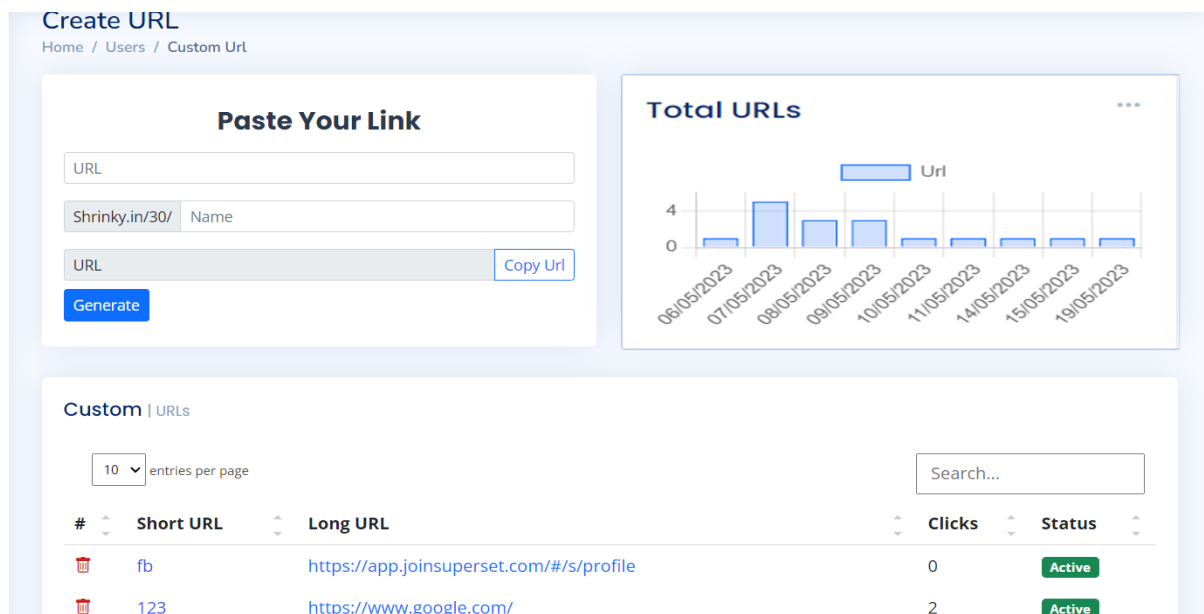


Fig 8.2.6 Custom section page

## QR Code:

The Fig:4.8 shows the visibility for authenticated user, in this module the user can generate a QR code for the shortened URL which when scanned, redirects to the original URL.

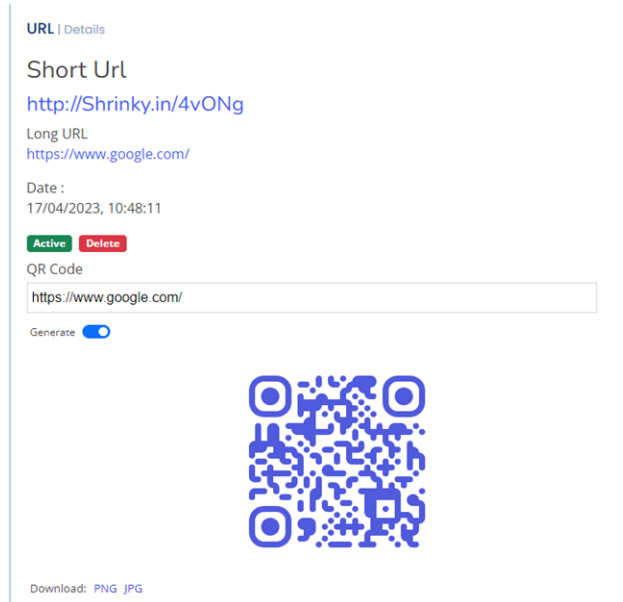


Fig 8.2.7 Generating QR Code

## Graph:

The Fig:4.9 shows the visibility for authenticated user, in this module the user can view the total number of URLs generated till date, total number of clicks on those URLs till date and total number of URLs used on that day.

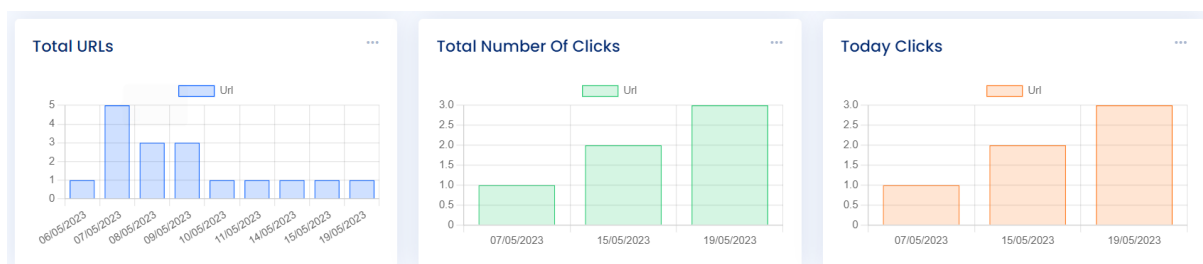


Fig 8.2.8 Graph

## **CHAPTER 9:**

### **CONCLUSION AND FUTURE SCOPE**

#### **9.1 CONCLUSION**

It is concluded that the application works well and satisfy the end users. This application is user friendly so everyone can use easily. We have done the requirement gathering and literature survey to short the URL and generate QR code for the shortened URL and fulfil the objectives. Proper documentation is provided. The end user can easily understand how the whole system is implemented by going through the documentation. Further enhancements can be made to the application, so that the application functions in a very attractive and useful manner than the present one.

#### **9.2 FUTURE SCOPE:**

There is always room for improvement in every system. VTU can send documents directly to a student's private account. Enable e-signing facility for the users so that they can digitally sign in their documents and share them with others. In future we can implement the process for checking if the provided URL is a Secure URL or Phishing URL.

## REFERENCES

- [1] Rohit Sankhala, Manan Kharbanda, Ankit Yadav, Pardeep Suthar, Parampreet Kaur, A URL Shortening Service Project in 2022.
- [2] [Coleman, J.](#) and . (2013), "Putting QR codes to the test", [New Library World](#), Vol. 114 No. 11/12, pp. 459-477. <https://doi.org/10.1108/NLW-05-2013-0044>
- [3] Michael A. Veronin, 'URL 404 File Not Found' in 2009.
- [4] Sandy J.J. Gould a,n , Anna L. Cox a , Duncan P. Brumby a , Sarah Wiseman, A systematic exploration of design trade-offs in link shortening services in 2016.
- [5] Daejin Choia,1 , Jinyoung Hanb, Selin Chuna, Efstratios Rapposc , Stephan Roberts , Ted Taekyoung Kwona, Bit.ly:Uncovering content publishing and sharing through URL shortening services in 2018.
- [6] M. Alajmi, I. Elashry, H. S. El-Sayed and O. S. Farag Allah, "Steganography of Encrypted Messages Inside Valid QR Codes," in IEEE Access, vol. 8, pp. 27861-27873, 2020, doi: 10.1109/ACCESS.2020.2971984.
- [7] Gibbs, M. (2009). URL shortening: \* A list of URL shorteners worth checking out. *Network World (Online)*, Retrieved from <https://www.proquest.com/trade-journals/url-shortening/docview/223738160/se-2>
- [8] Hyung-Jin, M., & Li, Y. (2017). Secure short URL generation method that recognizes risk of target URL. *Wireless Personal Communications*, 93(1), 269-283. doi: <https://doi.org/10.1007/s11277-016-3866-8>
- [9] Server status codes. (1998). *Active Server Developer's Journal*, 2(10), 15. Retrieved from <https://www.proquest.com/trade-journals/server-status-codes/docview/206872841/se-2>
- [10] Munro, J. (2004, Dec 28). Cut URLs down to size; long URLs can be unwieldy. here's how to make them easy to use.: The independent guide to IBM personal computers. *PC Magazine*, 23, 84. Retrieved from <https://www.proquest.com/magazines/cut-urls-down-size-long-can-be-unwieldy-heres-how/docview/203815144/se-2>
- [11] Nikiforakis, N., Maggi, F., Stringhini, G., Rafique, M.Z., Joosen, W., Kruegel, C., Piessens, F., Vigna, G., Zanero, S., 2014. Stranger danger: exploring the ecosystem of ad-based URL shortening services. In: Proceedings of the 23rd International Conference on World Wide Web Conference (WWW 2014).

- [12] Gupta, N., Aggarwal, A., Kumaraguru, P., 2014. bit.ly/malicious: Deep dive into short URL based e-crime detection. CoRR abs/1406.3687.
- [13] Newman, A., 2014. Bitly helps the red cross get to hope.ly – 2014
- [14] Alexander Neumann, Johannes Barnickel, Ulrike Meyer, Security and Privacy Implications of URL Shortening Services in 2011.
- [15] Chhabra, S., Aggarwal, A., Benevenuto, F., Kumaraguru, P., 2011. Phi.sh/social: The phishing landscape through short URLs. In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference.
- [16] Choi, D., Han, J., Chung, T., Ahn, Y.Y., Chun, B.G., Kwon, T.T., 2015. Characterizing conversation patterns in reddit: From the perspectives of content properties and user participation behaviours. In: Proceedings of the 2015 ACM on Conference on Online Social Networks (COSN 2015).
- [17] DGTraffic, 2012. Indonesia internet users. Online, accessed 20-Oct-2016. Gelley, B., John, A., 2015. Do i need to follow you? Examining the utility of the Pinterest follow mechanism. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work (CSCW 2014).
- [18] Goel, S., Watts, D.J., Goldstein, D.G., 2012. The structure of online diffusion networks. In: ACM Conference on Electronic Commerce (EC 2012).
- [19] Han, J., Choi, D., Choi, A.Y., Choi, J., Chung, T., Kwon, T.T., Rha, J.Y., Chuah, C.N., 2015. Sharing topics in Pinterest: Understanding content creation and diffusion behaviours. In: Proceedings of the 2015 ACM on Conference on Online Social Networks (COSN 2015).
- [20] Han, J., Choi, D., Chun, B.G., Kwon, T., Kim, H.C., Choi, Y., 2014. Collecting, organizing, and sharing pins in Pinterest: Interest-driven or social-driven? In: Proceedings of the 2014 ACM International Conference on Measurement and Modelling of Computer Systems (SIGMETRICS 2014).
- [21] Klien, F., Strohmaier, M., 2012. Short links under attack: geographical analysis of spam in a URL shortener network. In: Proceedings of the 23rd ACM Conference on Hypertext and social media (HT 2012).
- [22] Maggi, F., Frossi, A., Zanero, S., Stringhini, G., Stone-Gross, B., Kruegel, C., Vigna, G., 2013. Two years of short urls internet measurement: Security threats and countermeasures. In: Proceedings of the 22nd International Conference on World Wide Web (WWW 2013).



- [23] shortening services. In: Proceedings of the 23rd International Conference on World Wide Web Conference (WWW 2014).
- [24] Rodrigues, T., Benevenuto, F., Cha, M., Gummadi, K., Almeida, V., 2011. On word-of-mouth-based discovery of the web. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC 2011).
- [25] Wang, C., Ye, M., Huberman, B.A., 2012. From user comments to on-line conversations. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2012).
- [26] Wang, D., Navathe, S.B., Liu, L., Irani, D., Tamersoy, A., Pu, C., 2013. Click traffic analysis of short URL spam on twitter. In: Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Work-sharing.

## **ANNEXURE**

### **PUBLICATION:**

This paper is published in International Journal for Research In Applied Science and Engineering Technology Volume 11, Issue V, May-2023 ISSN (Online): 2321 – 9653, Estd 2013.