# MPTS-MODERN PERIODIC TABLE SAFARI

A PROJECT REPORT

Submitted By

**ARUN JOSEPH JACOB**

**Reg.No:SJC17MCA009**

to

the APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for
the award of the degree

*of*

**MASTER OF COMPUTER APPLICATIONS**

Department of Computer Science and Applications
**ST.JOSEPH'S COLLEGE OF ENGINERRING AND TECHNOLOGY**
Bharananganam, Pravithanam Road, Kottayam, Palai,
Choondacherry, Kerala 686579

**May, 2020**

# DECLARATION

I undersigned hereby declare that the project report “MPTS-Modern Periodic Table Safari”, submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bona fide work done by me under supervision of Mrs Rinu Mathew. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

**Place: Palai** **ARUN JOSEPH JACOB**

**Date:** **Reg.No: SJC17MCA009**

**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**
**ST.JOSEPH’S COLLEGE OF ENGINEERING AND TECHNOLOGY**
Palai, Choondacherry, Kottayam, 686579
(*Approved by AICTE and affiliated to APJ Abdul Kalam Technological University)*

# CERTIFICATE

This is to certify that the report entitled **“MPTS-MODERN PERIODIC TABLE SAFARI”** submitted by **“ARUN JOSEPH JACOB , Reg.No:SJC17MCA009”** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications is a bona fide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

**Mrs.Rinu Mathew**
Internal Guide

**Mr. Alex Jose**
Project Co-ordinator

**Dr.T.D.Jainendrakumar**
Head of the Department

*Viva-voce held on*:.................................................

**External Examiner 1:**

**External Examiner 2:**

# ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgment, then let words play the heralding role in expressing my gratitude. To bring something into existence is truly a work of God. I would like to thank God for not letting me down and showing me the silver lining in the dark clouds.

I would like to thank Dr. J David, Principal, St. Joseph’s College of Engineering & Technology for his support and encouragement. I convey my heartfelt thanks to Dr. T.D Jainendrakumar (Head of the Department - Master of Computer Applications, St. Joseph’s College of Engineering & Technology,) for providing an opportunity for the project presentation. It is my pleasure to express my gratitude to the project coordinator Mr. Alex Jose, Asst.professor, Department of Computer Applications, St. Joseph’s College of Engineering & Technology whose support and constructive criticism has led to the successful completion of the task.

With the biggest contribution to this report, I would like to thank Mrs. Rinu Mathew Asst.proffessor, Department of Computer science and Applications who had given me full support in guiding me with stimulating suggestions and encouragement to go ahead in all the time of the this work.

I would also thank my institution and faculty, my family and friends without whom this project would have been a distant reality.

**ARUN JOSEPH JACOB**

# ABSTRACT

As we all know the world is developing day by day with new technologies, new business etc. Only with the help of technologies the world can exist and also technologies can't exist only with computer science .They need a good partner I think Chemistry is a branch of science and also it is important in daily development of technologies. The basic of chemistry is elements and their properties. If you have a good understanding in elements you can easily crack chemistry. Periodic table also known as the periodic table of elements and it is a tabular display of elements arranged on the basis of atomic number, electronic configuration etc.

So think about the technology of periodic table in your hand. MPTS(MODERN PERIODIC TABLE SAFARI) is an app that you can easily crack elements by just one click. This project “Modern Periodic Table - Safari” is about to learn the periodic table through an android application. If a person wants to understand the periodic table elements, their features, uses, atomic weight, atomic number etc. He just needs to open this application. You want some come gets up. We are introducing the safari through periodic table with an android app.

System consist of 2 modules

- Admin
- Users

**Technology : Android**

**Database : Firebase**

# CONTENTS

# LIST OF TABLES

**IV1.5 Test Case**

# LIST OF FIGURES

# CHAPTER I

# INTRODUCTION

## 1.1 PROBLEM DEFINITION

As we all know the world is developing day by day with new technologies, new business etc. Only with the help of technologies the world can exist and also technologies can't exist only with computer science .They need a good partner I think Chemistry is a branch of science and also it is important in daily development of technologies. The basic of chemistry is elements and their properties. If you have a good understanding in elements you can easily crack chemistry. Periodic table also known as the periodic table of elements and it is a tabular display of elements arranged on the basis of atomic number, electronic configuration etc.

So think about the technology of periodic table in your hand. MPTS(MODERN PERIODIC TABLE SAFARI) is an app that you can easily crack elements by just one click. This project "Modern Periodic Table - Safari" is about to learn the periodic table through an android application. If a person wants to understand the periodic table elements, their features, uses, atomic weight, atomic number etc. He just needs to open this application. You want some come gets up. This is the safari through periodic table with an android app.

This application helps the users to view the periodic table, features of elements, uses of elements, organic and inorganic compounds. The users can submit their feedback. Users can also search for the specific elements. This application provides security by using the OTP system.

## I.2 ABOUT THE ORGANIZATION

The college was founded by a group of well known academics. They are pioneering educators, having unmatched experience in the field of education with a belief that the continuous search for knowledge is the sole path to success. The Primary focus of the institution is to expose the young minds to be world of technology, instilling in them confidence and fortitude to face new challenges that enables them to excel in their chosen field. The college inculcates the development of all facets of the mind culminating in an intellectual and balanced personality. Our team of dedicated and caring faculty strives to widen the students horizon of learning thereby achieving excellent results for every student. St. Joseph’s college of Engineering & Technology (SJCET), has always been in the forefront

with a wide spectrum of distinct features and facilities. The institution is a leader in the academia and its culture has set a benchmark in the region of quality in education. SJCET, Right from inception, has been maintaining high levels of standard in academic and extra curricular realms of activities. We offer BTEC degree courses in 6 engineering disciplines, and Masters Degree courses in Engineering, Computer Application and Business Administration. In the short span of a decade of its existence and among the six batches of students that have graduated, the college bagged several university ranks and has a remarkably high percentage of pass. The college is also the venue of national and state level seminars and symposiums and has emerged as the hub of technical education in the state. The placement scenario is also quite commendable, with several premier industries visiting ST. Joseph’s college of engineering & technology for placement and recruitment.

## I.3 OBJECTIVE OF THE PROJECT

“MPTS-Modern Periodic Table Safari” is created for fulfilling learning process of users. The basics of chemistry is elements and their properties. If you have a good understanding in elements you can easily crack chemistry. Periodic table also known as the periodic table of elements and it is a tabular display of elements arranged on the basis of atomic number, electronic configuration etc. This application helps the users to view the periodic table, features of elements, uses of elements, organic and inorganic compounds. The users can submit their feedback. Users can also search for the specific elements. This application provides security by using the OTP system.

# CHAPTER II

# LITERATURE SURVEY

## II.1 INITIAL INVESTIGATION

The purpose of the preliminary investigation is to determine whether the problem or deficiency in the current system really exists. The project developer may reexamine some of the feasibility aspects of the project. At this point, the purpose is to make a “go” or “no-go” decision. The end result is a decision to proceed further or to abandon the project. In the preliminary investigation an initial picture about the system working is got from this study and data collection methods are identified. To launch a system investigation we need a master plan detailing the steps to be taken, the people to be questioned and the outcome expected.

## II.2 EXISTING SYSTEM

Many chemistry related applications are designed for the learning process. But the students may suffer in finding the features and uses of each element in the periodic table. Searching is particular element is also a hard process in the existing systems. Users also suffer in finding the organic and inorganic compounds of each elements.

## II.3 PROPOSED SYSTEM

The proposed system is an application where the users can register and login to this application by using the OTP system. The users can easily find the period table elements. There is no suffer in searching the specific elements. The organic and inorganic compounds of each elements are available in the application. The users are provided with the features and uses of each elements in the periodic table. There is a facility to provide the feedback to the user.

### II.3.1 Advantages of the Proposed System

- The proposed system provides accurate data.
- An easily accessible, dynamic design.
- The proposed system is very much faster than existing issue
- Searching is taking small amount of computerized time.
- Attractive prominence & user friendly.
- Eliminate chances for errors and reduce effort

### II.3.2 Features of the Proposed System

- Easy editing, adding and deleting processes
- Provides more security than the existing system
- OTP system is used
- Easy searching
- Fast availability of periodic table structure
- Good validation checking

## II.4 FEASIBILITY STUDY

The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product. While evaluating the existing system, many advantages and disadvantages raised. Analyzing the problem thoroughly forms the vital part of the system study. Problematic areas are identified and information is collected. The main aim of feasibility study is to evaluate alternative site and propose the most feasible and desirable site for development. If there is no loss for the organization then the proposed system is considered financially feasible. A feasibility study is carried out to select the best system that meets performance requirements.

The benefits of this application are users can easily interact and get the services without much complexity. It helps to make it possible that more users can interact with the site at a time. Feasibility study is to determine whether the proposed system is technically, economically and behaviourally feasible in all aspects.

In this scenario, problems are identified. Essential data are being gathered for the existing problems. It is necessary that this analysis familiarizes the designer with objectives, activities, and the function of the organization in which the system is to be implemented. The feasibility study was divided into four:- Technical, Economical, Operational and Behavioural.

### II.4.1 TECHNICAL FEASIBILITY

According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs, are identified. While considering the problems of existing system, it is sufficient to implement the new system. The proposed system can be implemented to solve issues in the existing system. It includes the evaluation of and how it meets the proposed system. This system uses Firebase as database and Android technology

### II.4.2 ECONOMIC FEASIBILITY

Economic analysis is most frequent used for evaluating of the effectiveness of the candidate system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a candidate system and compare them with the existing system. Except for the initial capital amount and the amount after each financial year, no other huge amount is needed. The expenses can be handles by any participants. So, the system is economically feasible.

This feasibility involves some questions such as whether the firm can afford to build the system, whether its benefits should substantially exceed its costs, and whether the project has higher priority and profits than other projects that might use the same re- sources. Here there is no problem. This firm has fully equipped hard ware, and fully fledged software, so no need to spend money on these issues. And as the client and the developer are one, there is no further problem in economic issues.

### II.4.3 OPERATIONAL FEASIBILITY

Methods of processing and presentation are all according to the needs of clients since they can meet all user requirements here. The proposed system will not cause any problem under any circumstances and will work according to the specifications mentioned. Hence the proposed system is operationally feasible.

People are inherently resistant to change and computer has been known to facilitate changes. The system operation is the longest phase in the development life cycle of a system. So, Operational Feasibility should be given much importance. This system has a user-friendly interface. Thus it is easy to handle.

### II.4.4 BEHAVIORAL FEASIBILITY

In today’s world, computer is an inevitable entity. As per the definition of behaviour design, many valid points are recognized in this study. This system behaviour changes according to different environment. In order to ensure proper authentication and authorization and security of sensitive data of the admin or users, login facilities are provided. OTP system is also used. These are the main feasibility studies tested in this application.

# CHAPTER III
# SYSTEM ANALYSIS AND DESIGN

## III.1 SOFTWARE REQUIREMENT SPECIFICATION

The primary goal of the system analyst is to improve the efficiency of the existing system. For that study of specification of the requirement is very essential. For the development of the new system, a preliminary survey of the existing system will be conducted. An investigation is done whether the up gradation of the system into an application program could solve the problems and eradicate the inefficiency of the existing system. This gives an idea about the system specifications required to develop and install the project ”MPTS-Modern Periodic Table Safari”.

The System Requirements Specification is based on the System Definition. The requirement specifications are primarily concerned with functional and performance aspect of a software product and emphasis are placed on specifying product characteristics implying how the product will provide those characteristics. One of the most difficult tasks is selecting software, once the system requirement is find out then we have to determine whether a particular software package fits for those system requirements. This selection summarizes the application requirement.

### III.1.1 HARDWARE REQUIREMENT

- CPU – INTEL CORE i3
- HARD DISKSPACE - 500 GB
- RAM - 8GB
- DISPLAY - 19 STANDARD RATIO LCDMONITOR
- KEYBOARD - 99-104 KEYS
- CLOCK SPEED - 1.99 GHZ

### III.1.2 SOFTWARE REQUIREMENT

- OPERATING SYSTEM – UBUNTU
- IDE – ANDROID STUDIO

- FRONT END – XML
- BACK END –FIREBASE (Realtime database)

## III.2 SYSTEM DESIGN

Designing the system in an effective way leads to the smooth working of any software’s. System design is the process of developing specification for a candidate system that meet the criteria established in the system analysis. Major step in the system design is the preparation of the input forms and output reports in a form applicable to the user. The main objective of the system design is to use the package easily by any computer operator. System design is the creative act of invention, developing new inputs, and database, off-line files, method, procedure and output for processing business to meet an organization objective. System design builds information gathered during the system analysis. This system is designed neatly so that user will never get ambiguity while using the system.

## III.2.1 NON-FUNCTIONAL REQUIREMENTS

**Performance Requirements**

For the efficient performance of the application, network must have high bandwidth so that the task of centralized management does not lead to network jam. Also the hard disk capability must be high so that data can be effectively stored and retrieved.

**Security Requirements**

Security requirements of this application involves authentication using the mobile number. The OTP system used in this application will make more secure. There is a username and password verification during the login process so that invalid users are restricted from data access.

## III.3 UNIFIED MODELING LANGUAGE [UML]

UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar JAcobson and the RationalSoftware Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects. Today, UML is accepted by the Object Management Group (OMG) as the standard for modelling software development.

UML stands for Unified Modeling Language.UML 2.0 helps extend the original UML specification to cover a wider portion of software development efforts including agile practices.Improved integration between structural models like class diagrams and behavior models like activity diagrams.

The original UML specified nine diagrams; UML 2.x brings that number up to 13. The four new diagrams are called: communication diagram, composite diagram, interaction overview diagram and timing diagram. It also renamed state chart diagrams to state machine diagrams, also known as state diagrams.

**Types of UML diagrams**

The current UML standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing and deployment. These diagrams are organized into two distinct groups: structural diagrams and behavioral or interaction diagrams.

**Structural UML diagrams**

- Class diagram
- Package diagram

- Object diagram
- Component diagram
- Composite structure diagram
- Deployment diagram

**Behavioral UML diagrams**

- Activity Diagram
- Sequence diagram
- Use case diagram
- State diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

### III.3.1 Use case Diagram

To model a system the most important aspect is capture the dynamic behaviour. To modify a bit in details, dynamic behaviour of the system when it is running or operating. So only behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use case diagram consists of actors, use case and their relationships. The diagram is used to model the system of an application. A single use case diagram captures a particular functionality of a system.

Use case Diagram objects:

- Actor
- Use case
- System
- Package Actor

**Actor**

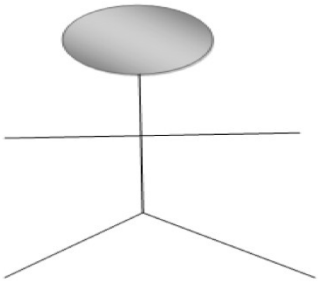


Figure III.1: Actor

Actor is a use case diagram in an entity that performs a role in one given system. This could be a person, organization or an external system usually drawn like skeleton.

### Use case

A use case represents a function or an action within the system. It’s drawn as an oval and named with the function.

Figure III.2: Use Case

### System

System is used to define the scope of the use case and drawn as a rectangle. This is an optional element but useful when your visualizing large systems. For example you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areas covered in different releases.

### Package

Package is another optional element that is extremely useful in complex diagrams. Similar to use class diagrams, packages are used to group together use cases.

The following is the UML diagram of this system:

I

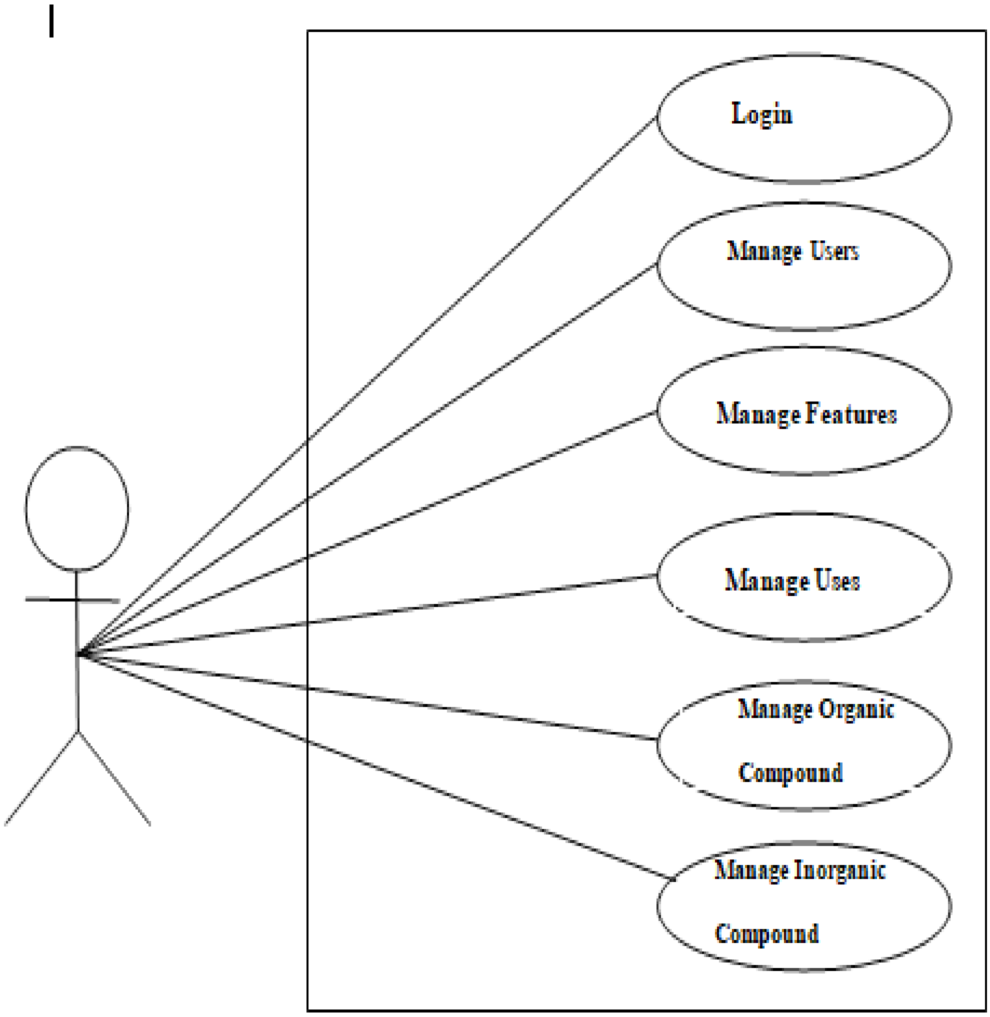


Figure III.3: UML DIAGRAM FOR ADMIN

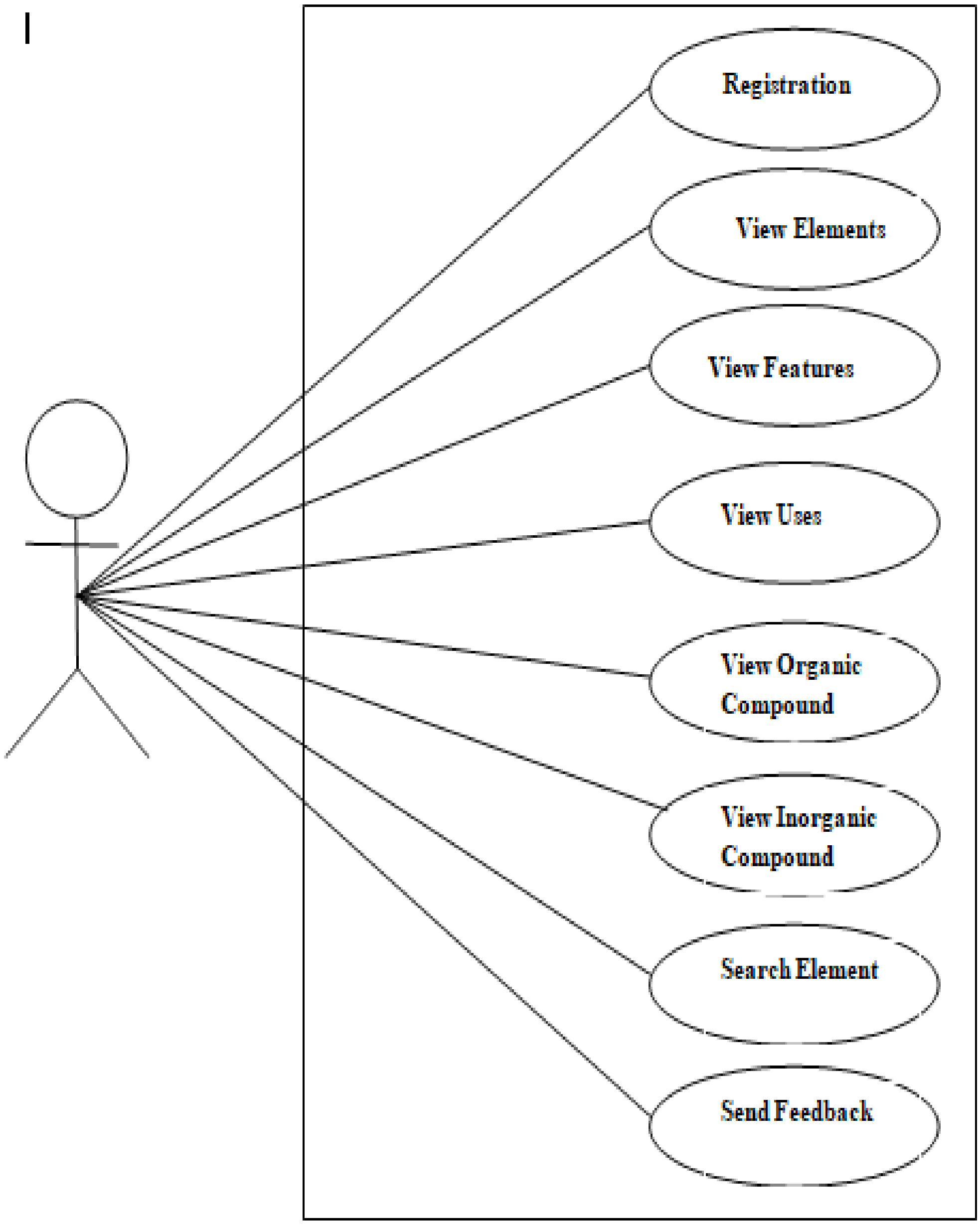


Figure III.4: UML DIAGRAM FOR USERS

Although UML sequence diagrams are typically used to describe object-oriented software systems, they are also extremely useful as system engineering tools to design system architectures in business process, as message sequence charts and call flows for telecoms or wireless system design, and for protocol stack design and analysis.

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. 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 typically associated with use case realizations in the logical View of the system under development. 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. This allows the specification of simple runtime scenarios in a graphical manner.

### III.3.2 Sequence Diagram

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interaction of the header elements.

Sequence Diagrams are used primarily to design, document and validate the archItecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

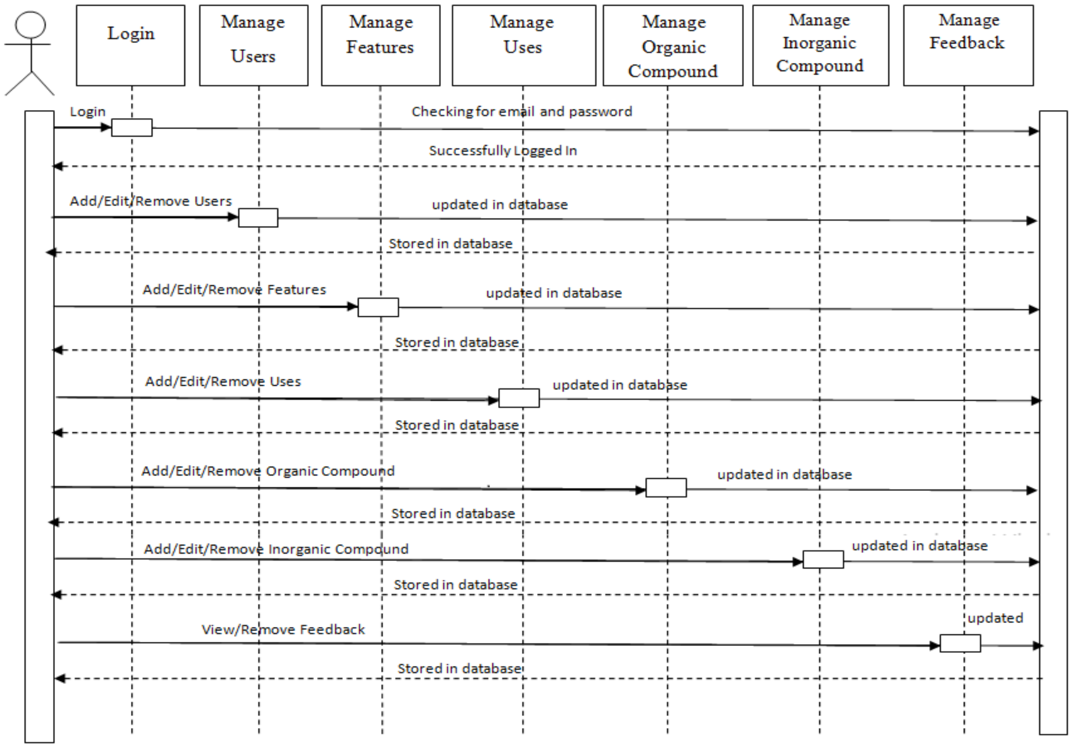


Figure III.5: Sequence Diagram For Admin

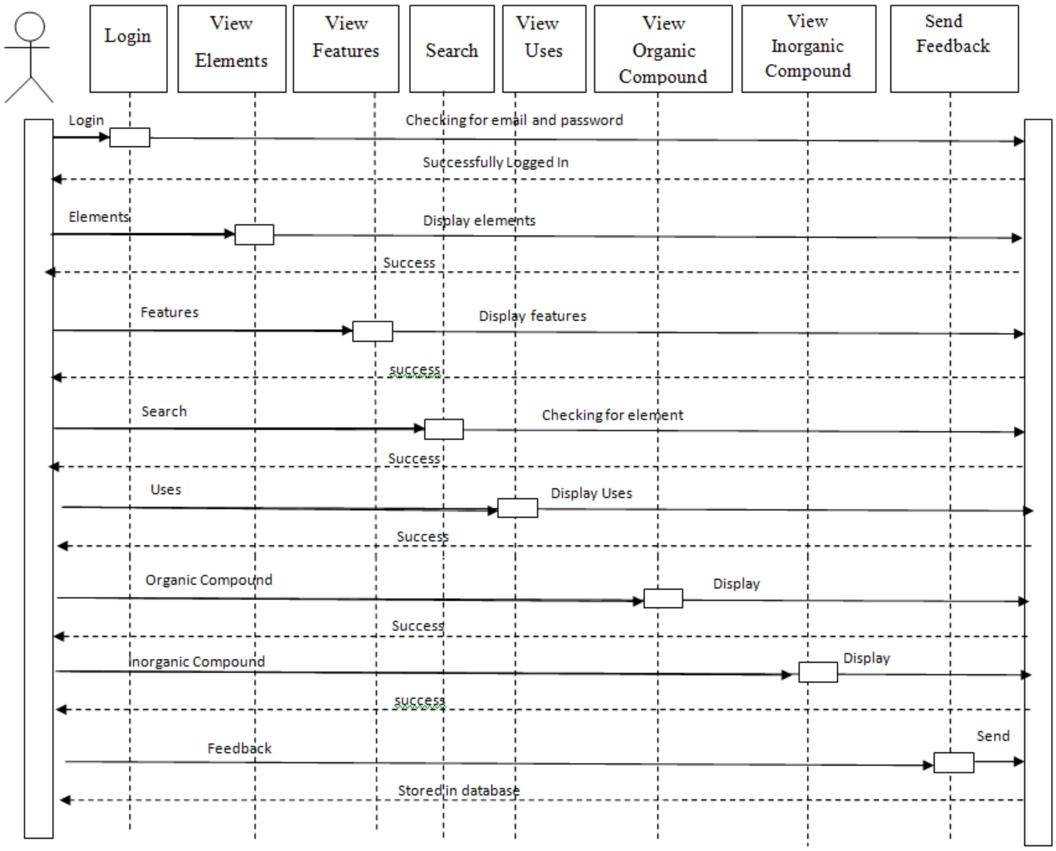


Figure III.6: Sequence Diagram For Users

### III.3.3 Activity Diagram

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

I

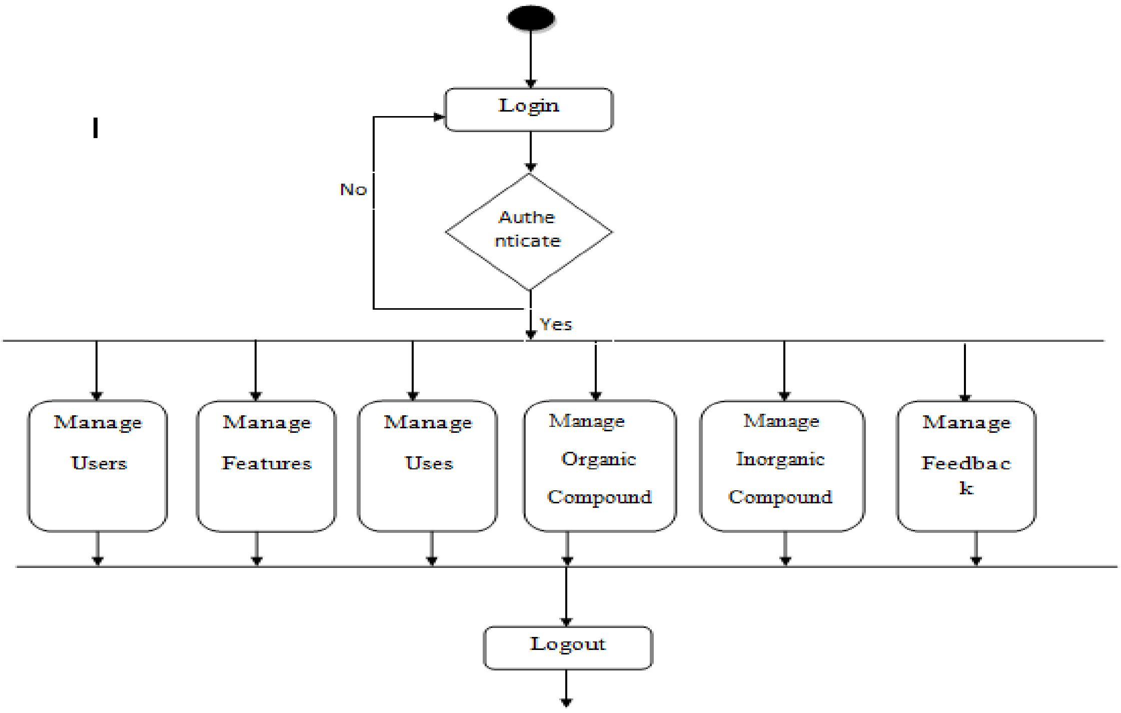


Figure III.7 Activity diagram- Admin

I

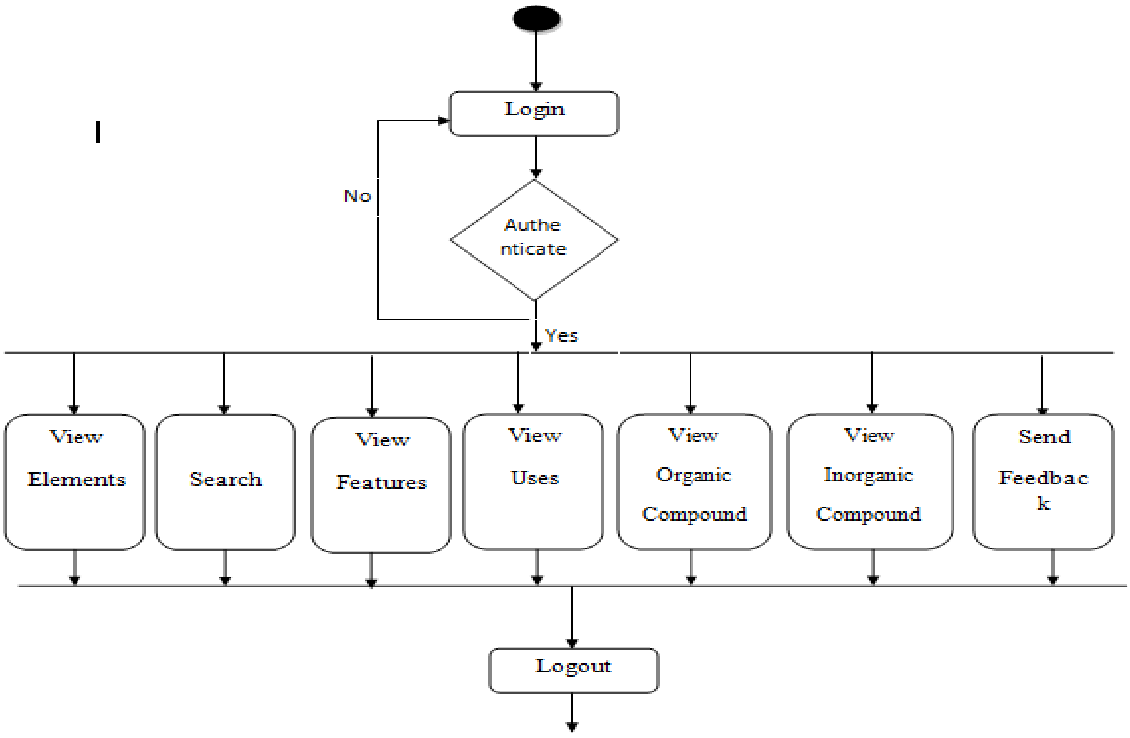


Figure III.8 Activity diagram-  Users

## III.4 SYSTEM DESIGN

The first step is to determine how the output is to be produced and in what for- mat. Then input data and master files have to be designed as the next step and finally the impact of the candidate system on the user and organization are documented and evaluated by the management. After identifying the problem and the limitation of the existing system, a detailed design of the proposed system is conducted.

Free flow personnel interview and reference to previous records prepared manually were the only methods taken to collect necessary information. At present, all organizations are on the path of computerization process.

Design is the phase that indicates the final system. It is the solution, the translation of requirements into ways of meeting them. In this phase the following elements were designed namely, data flow, data stores, processes, procedures was formulated in a manner that meet the project requirements. After logical design physical construction of the system is done.

The database tables, input screens, output screens, output reports are designed. Af- ter analyzing the various functions involved in the system the database, labels as dictionaries designed. Care is taken for the field name to be in self-explanatory form. Unnecessary fields are avoiding so as not affecting the storage system.

Care must be taken to design the input screen in the most user-friendly way so as to help even the novice users to make entries approximately in the right place. This is being accomplished by the use of giving online help messages, which are brief and cleanly prompts users for appropriate action. Design is the only way that we can accurately translate a customer’s requirements into a finished software product or system. Without design, risk of building an unstable system exist one that will fail when small changes are made, one that will be difficult to test.

All input screens in the system are user friendly and are designed in such a way that even a layman can operate. The sizes of all screens are standardized.

The importance of the software design can be stated with a single word quality. De- sign is a place where quality is fostered in software development. Design is the only way where requirements are actually translated into a finished software product or sys- tem.

## Mainly this project consists of 2 Modules:

- **Admin**
- **Users**

**Admin Module**

Administrator is the main actor in this system. Administrator can manage the whole application. Admin can view the users. Admin add element features, uses, organic compounds and inorganic compounds. Admin can view the feedbacks from users.

- **Admin Login**

By the Username and password admin can login to the system

- **Add/View/Edit/Remove Users**

Admin is can add, edit, view or delete users who are registered in the application MPTS

- **Add/View/Edit/Remove Features**

Admin is can add, edit, view or delete users features of each elements and this will be available for the users.

- **Add/View/Edit/Remove Uses**

Admin is can add, edit, view or delete uses  of each elements and this will be available for the users.

- **Add/View/Edit/Remove Organic compound**

Admin is can add, edit, view or delete organic compound of each elements and this will be available for the users.

- **Add/View/Edit/Remove Inorganic compound**

Admin is can add, edit, view or delete inorganic compound of each elements and this will be available for the users.

- **View Feedback**

Admin can view the feedbacks which are provided by the users.

**Users Module**

Users can register and login to this application. This application can be used by the users for learning purposes. Users can view elements, features, uses, organic compounds, inorganic compounds and send feedback to the admin.

- **User Login**

By the OTP send to the mobile number, users can login into the application.

- **View Features**

Users can view the features of the elements that are added by the admin.

- **View Uses**

Users can view the uses of the elements that are added by the admin.

- **View Organic Compound**

Users can view the organic compound of the elements that are added by the admin.

- **View Inorganic Compound**

Users can view the inorganic compound of the elements that are added by the admin.

- **Search Element**

Users can search for the elements

- **View Periodic table**

Users can view the periodic table

- **Send Feedback**

Users can send feedback to the admin.

## III.4.1 DATA FLOW DIAGRAM (DFD)

### Context Level DFD

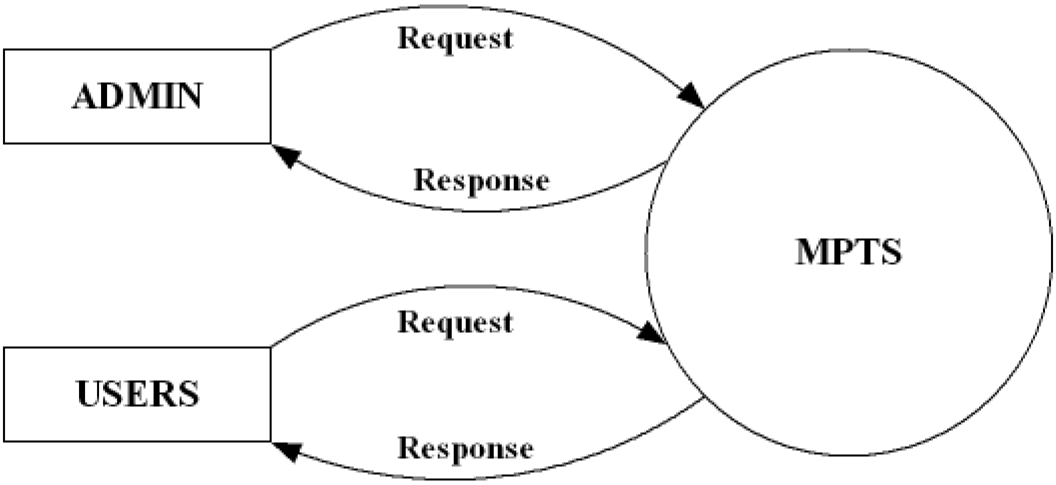


Figure III.9:Level 0

## Level 1 DFD- Admin

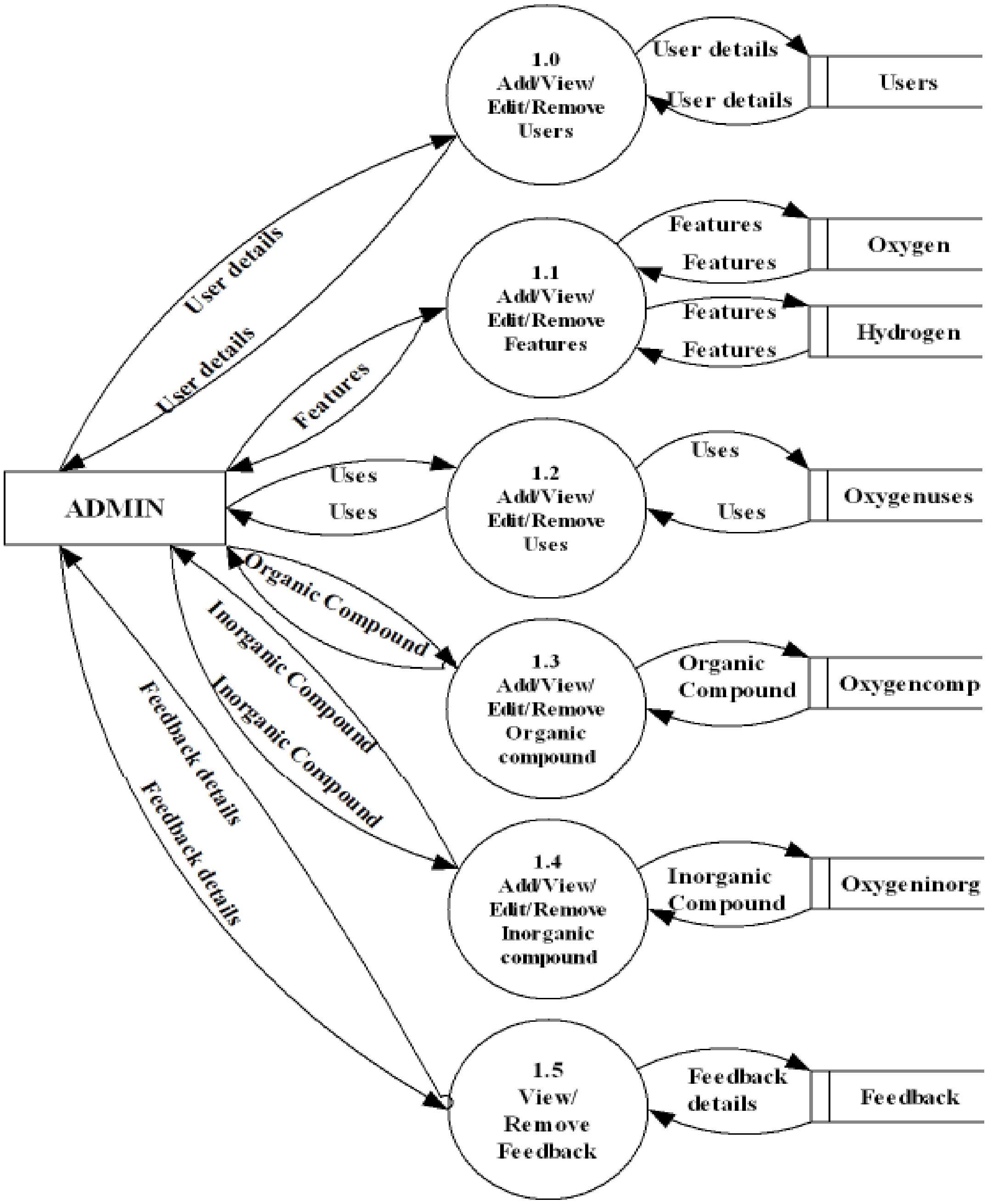


Figure III.10:Level 1 Admin

## Level 1 DFD- Users

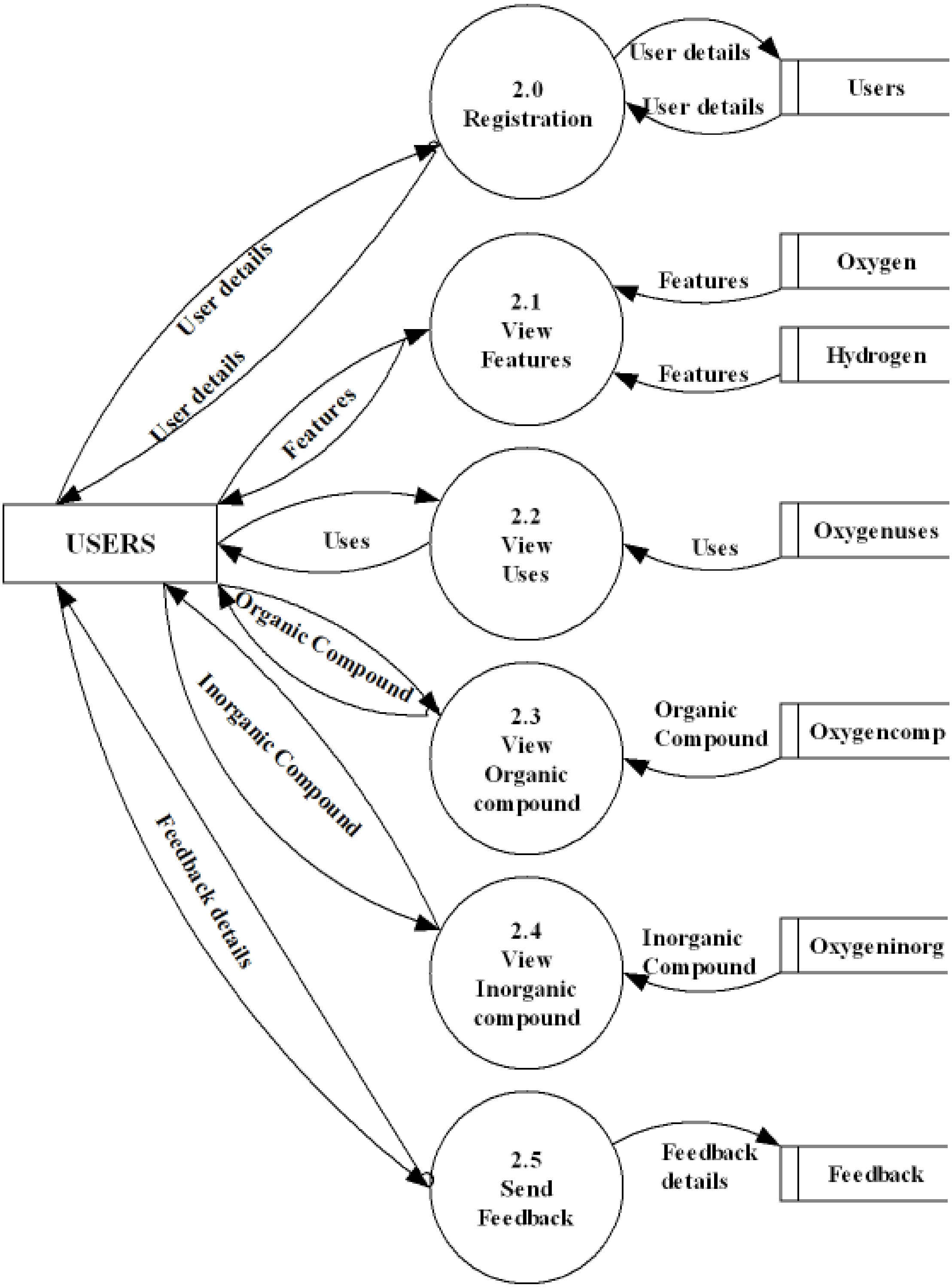


Figure III.11:Level 1 Users

### III.4.2 INPUT DESIGN

Input design is the process of converting user-oriented input into a computer based format. The goal of the designing input is to make data entry as easy and free from error. input to the system is entered through forms. A form is ”any surface on which information is to be entered, the nature of which is determined by what is already on that surface.” If the data going into the system is incorrect, then processing and output will magnify these errors. So designer should ensure that form is accessible and understandable by the user.

End-users are people who communicate to the system frequently through the user-interface , the design of the input screen should be according to their recommendations. The data is validated wherever it requires in the project. This ensures only correct data is entered to the system.GUI is the interface used in input design. All the input data are validated in the order and if any data violates any condition the use is warned by a message and asks to re-enter data. If the data satisfies all the conditions then it is transferred to the appropriate tables in the database. This project uses text boxes, list, buttons etc to accept user input. If user enters wrong format then it shows a message to the user.

### III.4.3 OUTPUT DESIGN

A quality output is one, which meets the requirement of the end user and presents the information clearly. In any system results of processing are communicated to the user and to the other systems through outputs. In the output design it is determined how the information is to be displayed for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship and helps user decision making.

It generally refers to the results and the information that are generated by the system. Effective, descriptive and useful design will improve the relationship with the user and the system because it is the direct source of information to the user. The objective of the output design is to convey the information of all the past activities, required status and to emphasize important events.

## III.4.4 TABLES

**1.User**

| **Field** | **Data type** | **Description** |
|---|---|---|
| user UID | String | User id |
| identifier | Integer | Mobile number |
| email id | String | Email address |
| Password | String | Password |
| Provider | String | Sign in method |
| created | Date | Date of creation |
| Signed in | Date | Date of sign in |

Table III.1: Table User

**2.Features**

| **Field** | **Data type** | **Description** |
|---|---|---|
| featureid | String | Feature id |
| features | String | Features of elements |

Table III.2: Table  Features

**3.Uses**

| **Field** | **Data type** | **Description** |
|---|---|---|
| useid | String | User id |
| uses | String | Uses of elements |

Table III.3: Table Uses

**4.Comp**

| **Field** | **Data type** | **Description** |
|---|---|---|
| organicid | String | Organic id |
| organiccompund | String | Organic compounds of elements |

Table III.4: Table Comp

**5.Inorg**

| **Field** | **Data type** | **Description** |
|---|---|---|
| inorganicid | String | Inorganic id |
| inorganiccompund | String | Inorganic compounds of elements |

Table III.5: Table Inorg

**6.Feedback**

| **Field** | **Data type** | **Description** |
|---|---|---|
| feedbackid | String | Feedback id |
| fd | String | Feedback |

Table III.6: Table Feedback

## III.5 TOOLS AND PLATFORMS

### III.5.1 Android

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google

The Android operating system (OS) is based on the Linux kernel. Unlike Apple's iOS, Android is open source, meaning developers can modify and customize the OS for each phone. Therefore, different Android-based phones often have different graphical user interfaces GUIs even though they use the same OS.

Android phones typically come with several built-in applications and also support third-party programs. Developers can create programs for Android using the free Android software developer kit (SDK). Android programs are written in Java and run through a Java virtual machine JVM that is optimized for mobile devices. The "Dalvik" JVM was used through

Android 4.4 and was replaced by Android Runtime or "ART" in Android 5.0. Users can download and install Android apps from Google Play and other locations.

### III.5.2 Android Software Development

Android software development is the process by which new applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible. All non-JVM languages, such as Go, JavaScript, C, C++ or assembly, need the help of JVM language code, that may be supplied by tools, likely with restricted API support. Some programming languages and tools allow cross-platform app support (i.e. for both Android and iOS). Third party tools, development environments, and language support have also continued to evolve and expand since the initial SDK was released in 2008. In addition, with major business entities like Walmart, Amazon, and Bank of America eyeing to engage and sell through mobiles, mobile application development is witnessing a transformation.

### III.5.3 Android SDK

The Android SDK (software development kit) is a set of development tools used to develop applications for Android platform. The Android SDK includes the following:

- Required libraries
- Debugger
- An emulator
- Relevant documentation for the Android application program interfaces (APIs)
- Sample source code
- Tutorials for the Android OS

Every time Google releases a new version of Android, a corresponding SDK is also released. To be able to write programs with the latest features, developers must download and install each version’s SDK for the particular phone.

The development platforms that are compatible with SDK include operating systems like Windows (XP or later), Linux (any recent Linux distribution) and Mac OS X (10.4.9 or later). The components of Android SDK can be downloaded separately. Third party add-ons are also available for download.

Although the SDK can be used to write Android programs in the command prompt, the most common method is by using an integrated development environment (IDE). The recommended IDE is Eclipse with the Android Development Tools (ADT) plug-in. However, other IDEs, such as NetBeans or IntelliJ, will also work. Most of these IDEs provide a graphical interface enabling developers to perform development tasks faster. Since Android applications are written in Java code, a user should have the Java Development Kit (JDK) installed.

### III.5.4 Android Debug Bridge (adb)

Android Debug Bridge (adb) is a versatile command-line tool that lets you communicate with a device. The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a Unix shell that you can use to run a variety of commands on a device. It is a client-server program that includes three components:

A client, which sends commands. The client runs on your development machine. You can invoke a client from a command-line terminal by issuing an adb command.

A daemon (adbd), which runs commands on a device. The daemon runs as a background process on each device.

A server, which manages communication between the client and the daemon. The server runs as a background process on your development machine.

adb is included in the Android SDK Platform-Tools package. You can download this package with the SDK Manager, which installs it at android_sdk/platform-tools/. Or if you want the standalone Android SDK Platform-Tools package, you can download it here.

For information on connecting a device for use over ADB, including how to use the Connection Assistant to troubleshoot common problems, see Run apps on a hardware device.

When you start an adb client, the client first checks whether there is an adb server process already running. If there isn't, it starts the server process. When the server starts, it binds to local TCP port 5037 and listens for commands sent from adb clients—all adb clients use port 5037 to communicate with the adb server.

The server then sets up connections to all running devices. It locates emulators by scanning odd-numbered ports in the range 5555 to 5585, the range used by the first 16 emulators. Where the server finds an adb daemon (adbd), it sets up a connection to that port. Note that each emulator uses a pair of sequential ports — an even-numbered port for console connections and an odd-numbered port for adb connections. For example:

Emulator 1, console: 5554

Emulator 1, adb: 5555

Emulator 2, console: 5556

Emulator 2, adb: 5557

and so on...

As shown, the emulator connected to adb on port 5555 is the same as the emulator whose console listens on port 5554.

Once the server has set up connections to all devices, you can use adb commands to access those devices. Because the server manages connections to devices and handles commands from multiple adb clients, you can control any device from any client (or from a script).

To use adb with a device connected over USB, you must enable USB debugging in the device system settings, under Developer options.

On Android 4.2 and higher, the Developer options screen is hidden by default. To make it visible, go to Settings > About phone and tap Build number seven times. Return to the previous screen to find Developer options at the bottom.On some devices, the Developer options screen might be located or named differently.

You can now connect your device with USB. You can verify that your device is connected by executing adb devices from the android_sdk/platform-tools/ directory. If connected, you'll see the device name listed as a "device."

### III.5.5 Android NDK

Code written in C/C++ can be compiled to ARM, or x86 native code (or their 64-bit variants) using the Android Native Development Kit (NDK). The NDK uses the Clang compiler to compile C/C++. GCC was included until NDK r17, but removed in r18 in 2018.

Native libraries can be called from Java code running under the Android Runtime using System.loadLibrary, part of the standard Android Java classes. Command-line tools can be compiled with the NDK and installed using adb.

Android uses Bionic as its C library, and the LLVM libc++ as its C++ Standard Library. The NDK also includes a variety of other APIs: zlib compression, OpenGL ES or Vulkan graphics, OpenSL ES audio, and various Android-specific APIs for things like logging, access to cameras, or accelerating neural networks.

The NDK includes support for CMake and its own ndk-build (based on GNU Make). Android Studio supports running either of these from Gradle. Other third-party tools allow integrating the NDK into Eclipse and Visual Studio.

For CPU profiling, the NDK also includes simpleperf which is similar to the Linux perf tool, but with better support for Android and specifically for mixed Java/C++ stacks.

### III.5.6 Firebase

Firebase is a mobile platform that helps you quickly develop high-quality apps, grow your user base, and earn more money. Firebase is made up of complementary features that you can mix-and-match to fit your needs, with Google Analytics for Firebase at the core. Firebase allows you to build apps which need authentication, database, file storage, analytics and server side functionality without having to own and manage infrastructure and software required for server side support. Firebase Realtime Database is a cloud-hosted database that supports multiple platforms Android, iOS and Web. All the data is stored in JSON format and any changes in the data, reflects immediately by performing sync across all the platforms & devices.

Following are the advantages of using Firebase Realtime Database:

**Realtime:** The data stored in the Firebase Realtime Database will be reflected at realtime i.e. if there is a change in the values in the database then that change will be reflected back to all the users at that instant only and no dealy will be there.

**Large Accessibility:** The Firebase Realtime Database can be accessed from various platforms like Android, iOS, Web. So, you need not write the same code for different platforms a number of times.

**Offline Mode:** This is the best advantage of using Firebase Realtime Database. If you are not connected with the internet and you changed something on your application then that change will be reflected in your application at that time only but on the Firebase Database, the change will be updated once you are online i.e. your device is connected to the internet. So, even if there is no internet, the user feels like using the services the same as done when there is the internet.

**No Application Server:** There is no need for application server here because the data is directly accessed from the mobile device.

**Control access to data:** By default, no one is allowed to change the data in the Firebase Realtime Database but you can control the access of data i.e. you can set which user can access the data.

# CHAPTER IV
# SYSTEM TESTING

## IV.1 TESTING METHODOLOGIES AND STRATEGIES

Software testing is an integral part of to ensure software quality, some software organizations are reluctant to include testing in their software cycle, because they are afraid of the high cost associated with the software testing .There are several factors that attribute the cost of software testing. Creating and maintaining large number of test cases is a time consuming process. Furthermore, it requires skilled and experienced testers to develop great quality test cases. . Testing begins at the module level and work towards the integration of entire computer based system. No testing is completed without verification and validation part.

The goal of verification and validation activities are to access and improve the quality of work products generated during the development and modification of the software. Testing plays a vital role in determining the reliability and efficiency of the software and hence is very important stage in software development. Tests are to be conducted on the software to evaluate its performance under a number of conditions. Ideally, it should do so at the level of each module and also when all of them are integrated to form the completed system.

### IV.1.1 Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

**Black Box Testing -** Using which the user interface, input anThe purpose is to validate that each unit of the software performs as designed.d output are tested.

**White Box Testing** - used to test each one of those functions behaviour is tested.

**Gray Box Testing** - Used to execute tests, risks and assessment methods

### IV.1.2 Integration Testing

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

After splitting the programs into units, the units were tested together to see the defects between each module and function. It is testing to one or more mod- ules or functions together with the intent of finding interface defects between the modules or functions. Testing completed at as part of unit or functional testing, integration testing can involve putting together of groups of modules and functions with the goal of completing and verifying meets the system requirements.

### IV.1.3 System Testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.The perquisites for System Testing are:-– All the components should have been successfully Unit Tested.

– All the components should have been successfully integrated.

– Testing should be completed in an environment closely resembling the production environment. When necessary iterations of System Testing are done in multiple environments.

## IV1.4 User Acceptance Testing

The system was tested by a small client community to see if the program met the requirements defined the analysis stage. It was fond to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in the analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system.

## IV1.5 Test Case

### 1.5.1 Test case for login activity.

| **Test Step** | **Expected Result** | **Actual Result** | **Status** |
|---|---|---|---|
| Checking the already entered number and country code. | Message like “Checking..” should appear. | The page is navigated to the home activity. | Pass |

### 1.5.2 Test case for OTP activity.

| Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|
| Select country code,enter phone number and click the next button. | Message like “Sending OTP...” should appear | A message like “phone verified” with id and go to the register activity | Pass |
| Enter the field with invalid phone number . | Message like “Valid Phone Required” should appear. | A message “Valid Phone Required” appears | Pass |
| Entered phone number didn’t get OTP. | Message like “OTP Timeout, Please Regenerate the OTP Again” should appear | A message “OTP Timeout, Please Regenerate the OTP Again” appears | Pass |

### 1.5.2 Test case for Registration activity

| Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|
| Enter all fields with valid entries and click“SAVE” button. | Message like “onSuccess :User Profile Created” and go to the home activity. | A messsage “onSuccess :User Profile Created” appears and go to the home activity. | Pass |
| Enter fields with invalid entries. | Message like “onFailure:Failed to Create User profile” should appear. | A messsage “onFailure: Failed to Create User Profile” appears | Pass |
| No fields are filled but click “SAVE” button. | Message like “Fill the required Details” should appear. | A messsage “Fill the required Details” appears. | Pass |

### 1.5.3 Test case for Feedback activity.

| **Test Step** | **Expected Result** | **Actual Result** | **Status** |
|---|---|---|---|
| Enter feedback in the field and click “SEND”button. | Message like “Your Feedback is sent” should appear. | A message “Your Feedback is sent” appears. | Pass |

# CHAPTER V
# SYSTEM IMPLEMENTATION

The primary goal of implementation is to write the source code to its specification that can be achieved by making the source code clear and straight forward as possible. Implementation means the process of converting a new or revised system design into operational one. The implementation is one phase of software development. Implementation is that stage in the project where theoretical design is turned into working system. Implementation involves placing the complete and tested software system into actual work environment. Implementation is concerned with translating design specification with source code. The three types of implementation are:-implementation of a computerized system to replace a manual system, implementation of a new system to replace existing one and implementation of a modified system to replace an existing one.

The implementation is the final stage and it is an important phase. It involves the individual programming; system testing, user training, and the operational running of developed proposed system that constitute the application subsystem. The implementation phase of the software development is concerned with translating design specification in the source code. The user tests the developed system and the changes are according to the needs. Before implementation, several tests have been conducted to ensure no errors encountered during the operation. The implementation phase ends with an evaluation of the system after placing it into operation of time. The validity and proper functionality of all the modules of the developed application is assured during the process of implementation. Implementation is the process of assuring that the information system is operational and then allowing user to take over its operation for use and evaluation. Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operated the new system. The most crucial stage in achieving a new successful system is that it works effectively and efficiently.

# CHAPTER VI
# CONCLUSION

“MPTS-Modern Periodic Table Safari” is created for fulfilling learning process of users. The basic of chemistry is elements and their properties. If you have a good understanding in elements you can easily crack chemistry. Periodic table also known as the periodic table of elements and it is a tabular display of elements arranged on the basis of atomic number, electronic configuration etc. This application helps the users to view the periodic table, features of elements, uses of elements, organic and inorganic compounds. The users can submit their feedback. Users can also search for the specific elements. This application provides security by using the OTP system.

The users can easily find the period table elements. There is no suffer in searching the specific elements. The organic and inorganic compounds of each elements are available in the application. The users are provided with the features and uses of each elements in the periodic table. There is a facility to provide the feedback to the user.

This project “MPTS - Modern Periodic Table Safari” is about to learn the periodic table through an android application. If a person wants to understand the periodic table elements, their features, uses, atomic weight, atomic number etc. He just needs to open this application. You want some come gets up. We are introducing the safari through periodic table with an android app.

# REFERENCES

1. https://developer.android.com/studio
2. https://developer.android.com/docs
3. https://console.firebase.google.com/
4. https://www.w3schools.com/
5. https://stackoverflow.com/

# APPENDIX A
# APPENDICES

## A.1 SCREEN SHOTS INPUT FORM, OUTPUT FORMS

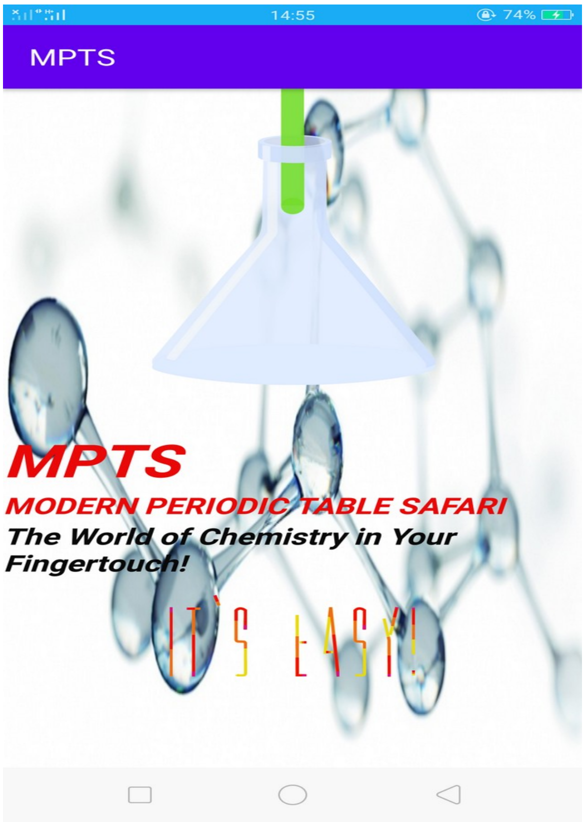
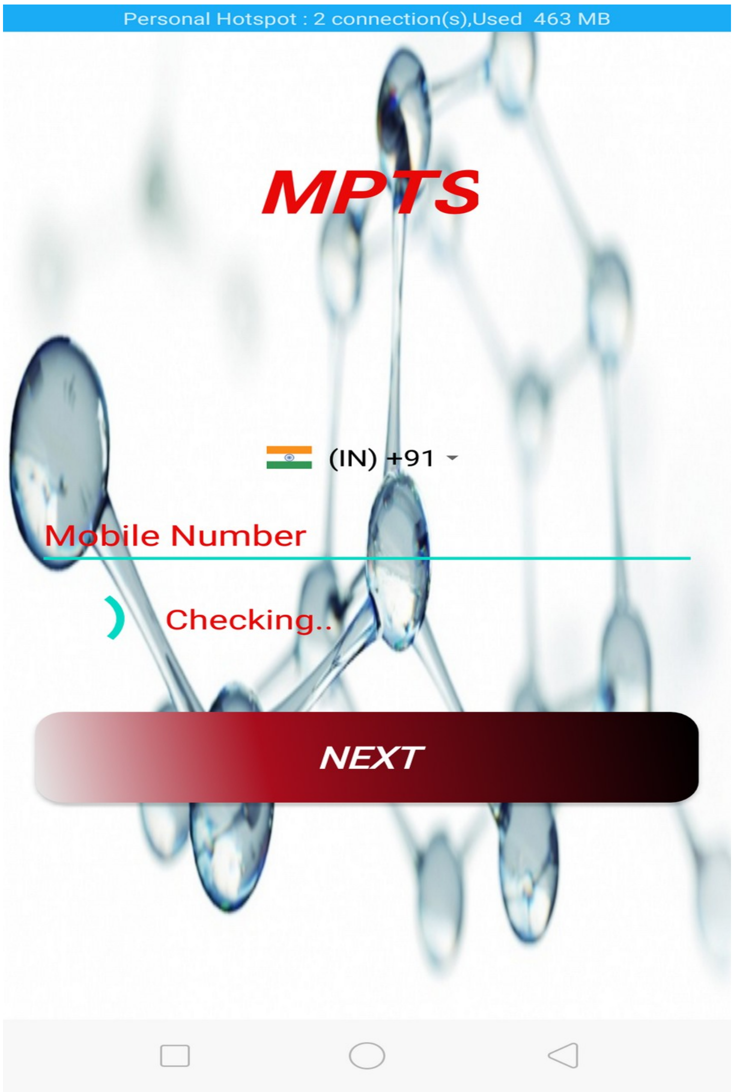

14:55
74%
MPTS
MPTS
MODERN PERIODIC TABLE SAFARI
The World of Chemistry in Your Fingertouch!
IT`S EASY!

Figure A.1  Front  Activity



Personal Hotspot : 2 connection(s),Used 463 MB

MPTS

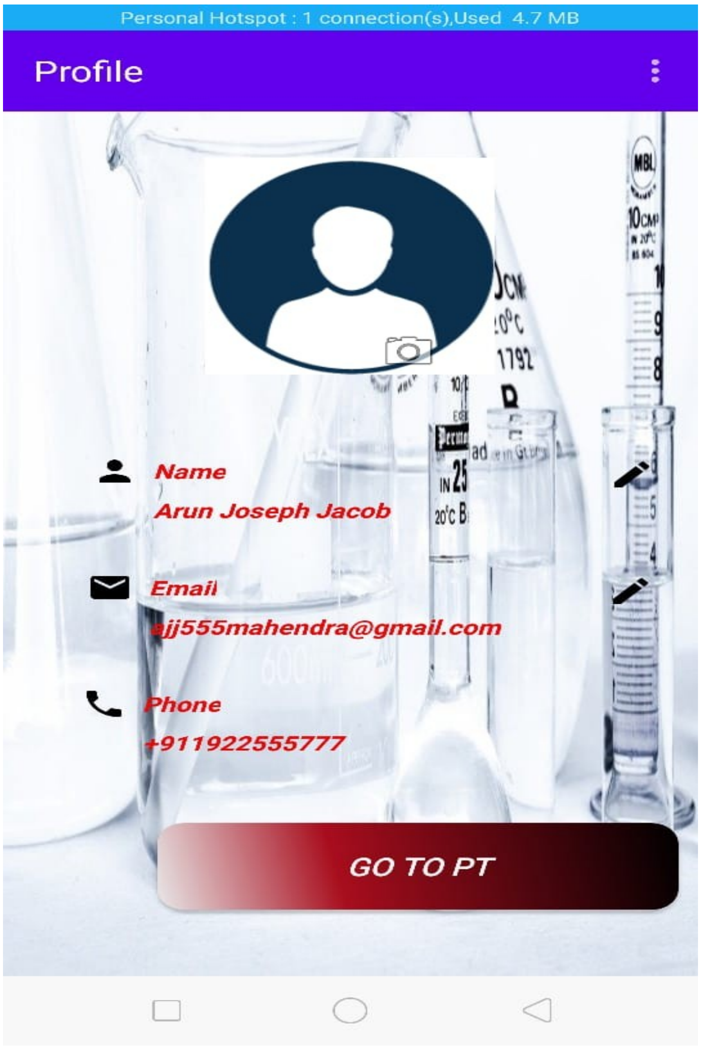
(IN) +91

Mobile Number

Checking..

Figure A.2: Login Activity

Figure A.3: Profile Activity

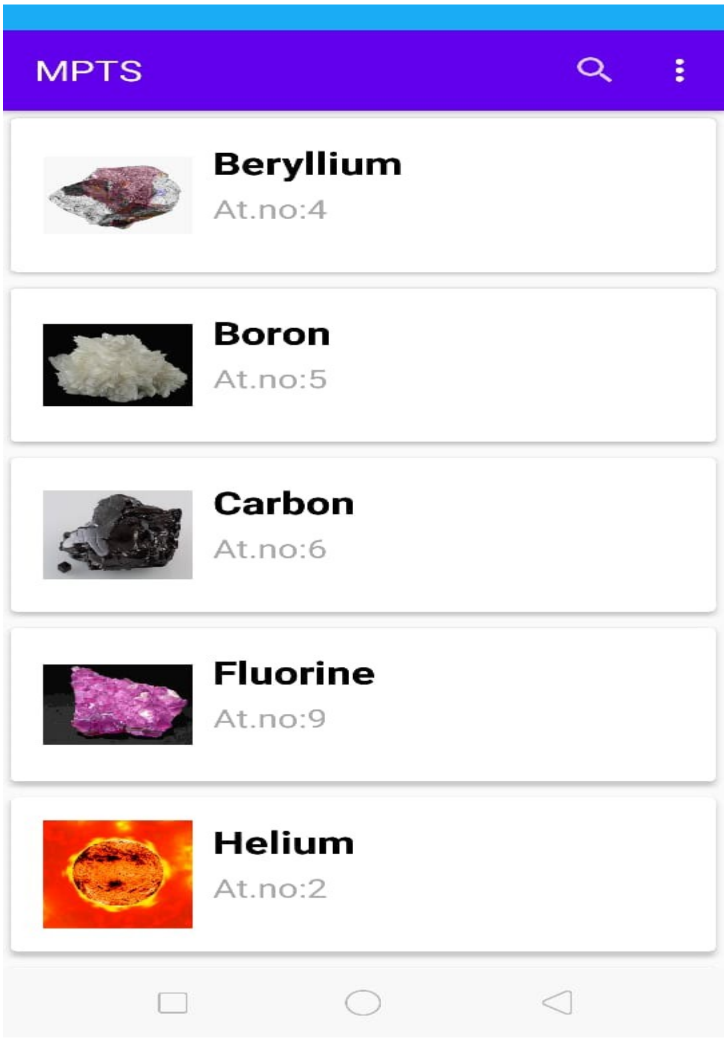


Name
Arun Joseph Jacob

Email
ajj555mahendra@gmail.com

Phone
+911922555777

Figure A.4 View Elements



MPTS

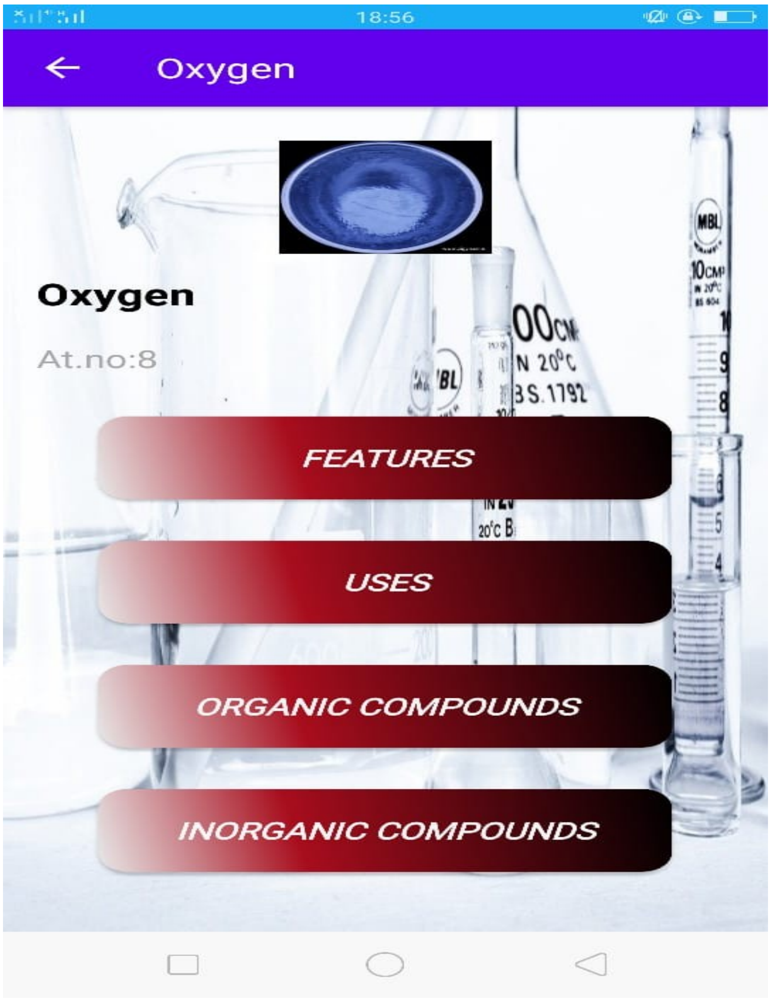
Beryllium
At.no:4

Boron
At.no:5

Carbon
At.no:6

Fluorine
At.no:9

Helium
At.no:2



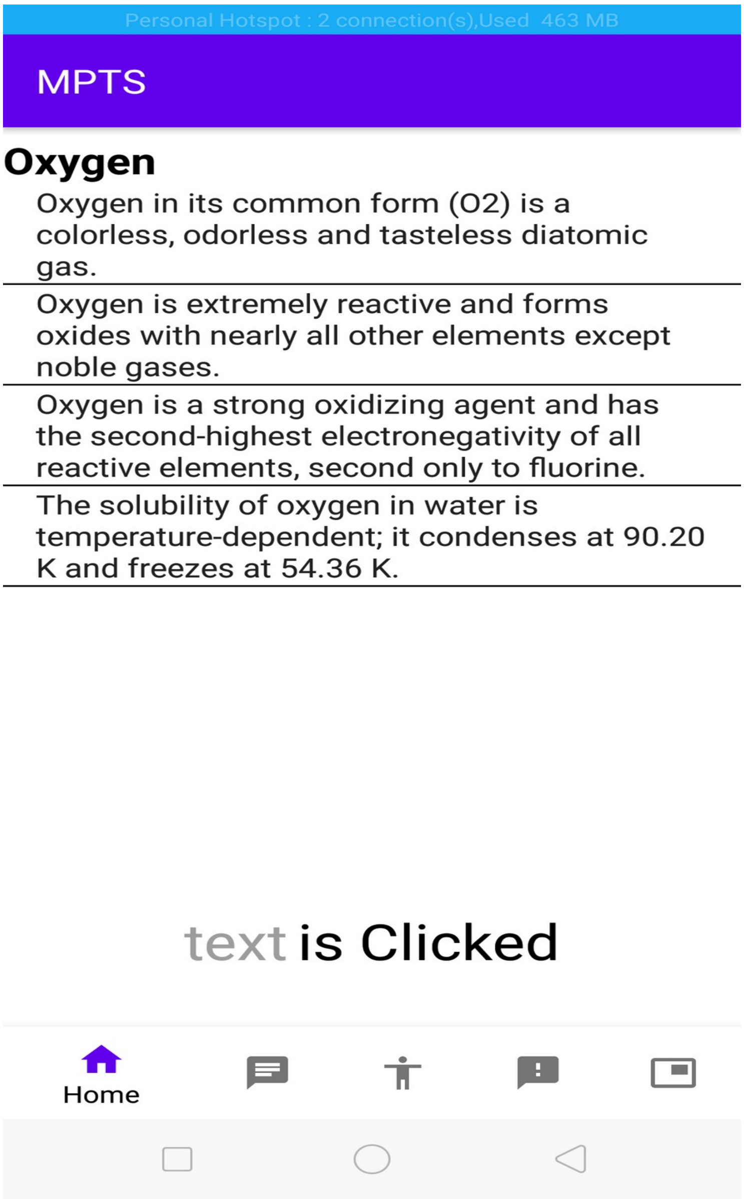
18:56

Oxygen

Oxygen

At.no:8

Figure A.5 Element description  Activity



Oxygen

Oxygen in its common form (O2) is a colorless, odorless and tasteless diatomic gas.

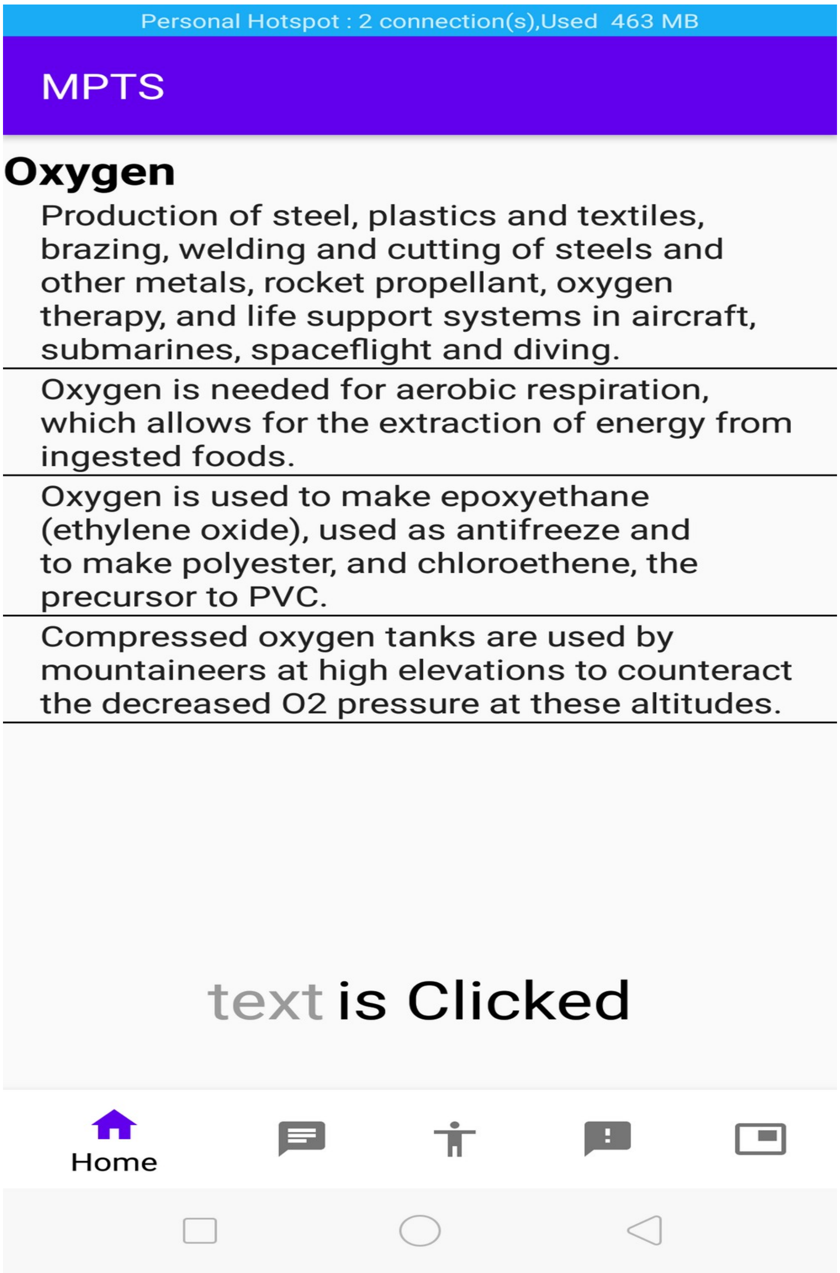
Oxygen is extremely reactive and forms oxides with nearly all other elements except noble gases.

Oxygen is a strong oxidizing agent and has the second-highest electronegativity of all reactive elements, second only to fluorine.

The solubility of oxygen in water is temperature-dependent; it condenses at 90.20 K and freezes at 54.36 K.

text is Clicked

Figure A.6 Element Features



**Oxygen**

Production of steel, plastics and textiles, brazing, welding and cutting of steels and other metals, rocket propellant, oxygen therapy, and life support systems in aircraft, submarines, spaceflight and diving.

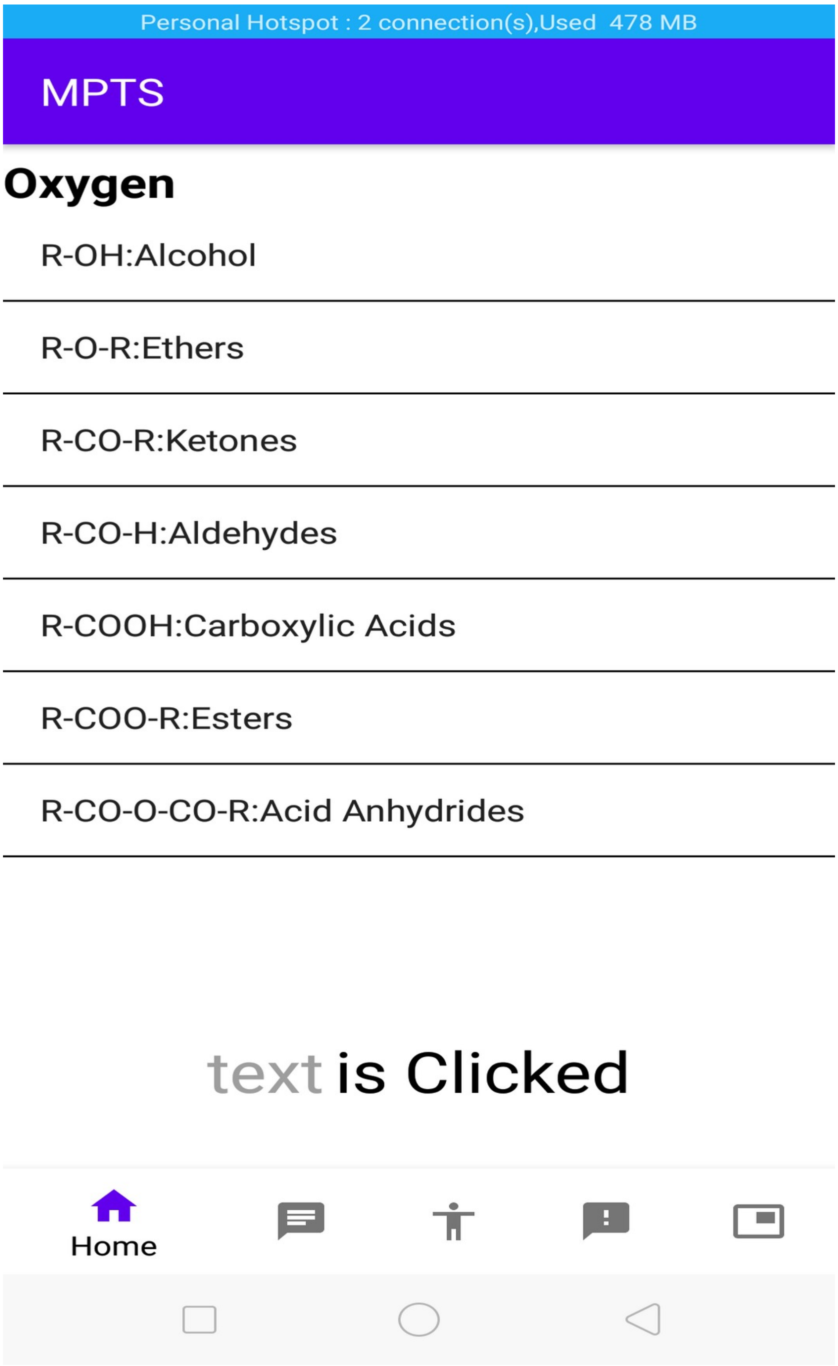
Oxygen is needed for aerobic respiration, which allows for the extraction of energy from ingested foods.

Oxygen is used to make epoxyethane (ethylene oxide), used as antifreeze and to make polyester, and chloroethene, the precursor to PVC.

Compressed oxygen tanks are used by mountaineers at high elevations to counteract the decreased O2 pressure at these altitudes.

text is Clicked

Figure A.7: Element Uses



Personal Hotspot : 2 connection(s),Used 478 MB

MPTS

Oxygen

R-OH:Alcohol

R-O-R:Ethers

R-CO-R:Ketones

R-CO-H:Aldehydes

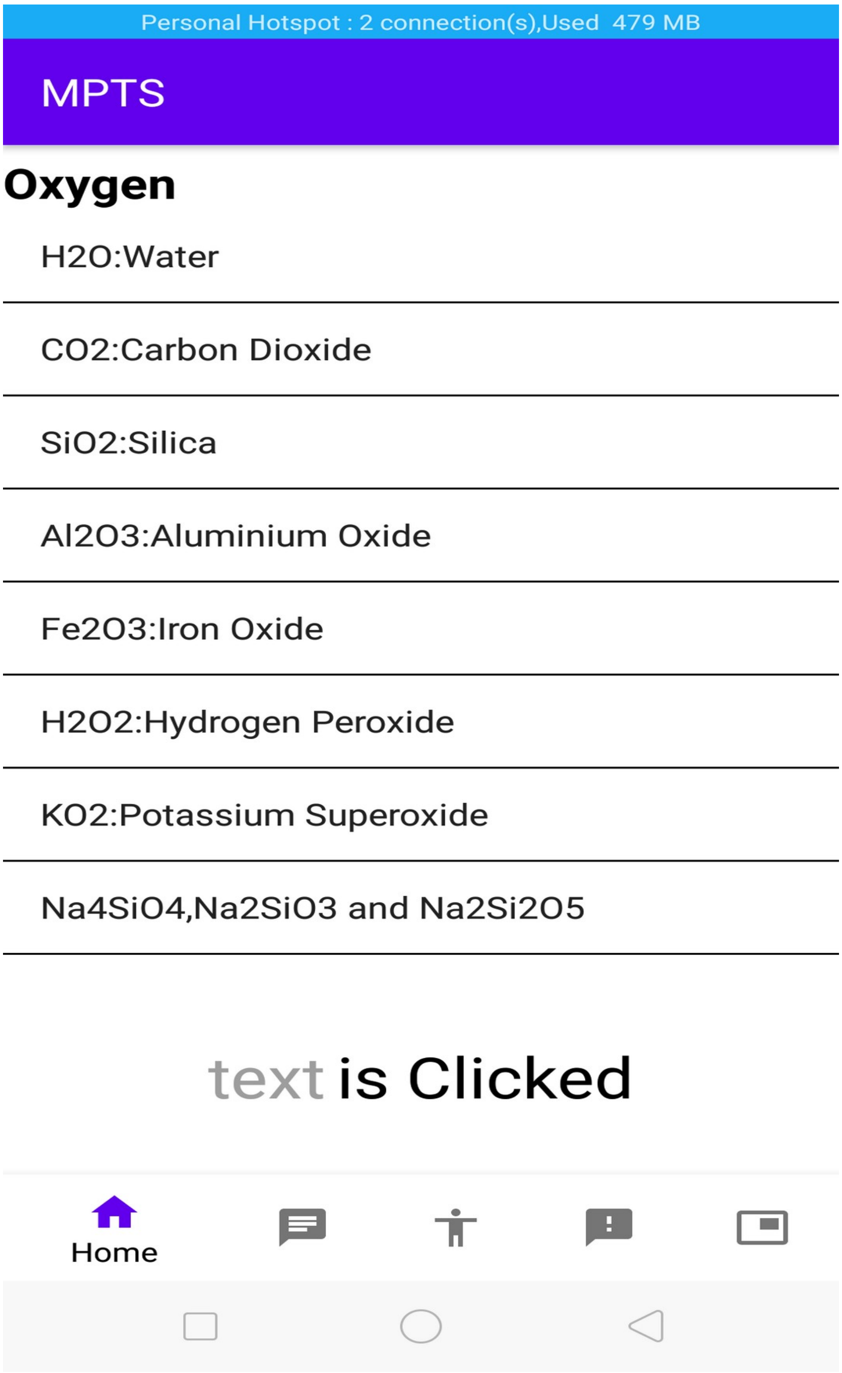
R-COOH:Carboxylic Acids

R-COO-R:Esters

R-CO-O-CO-R:Acid Anhydrides

text is Clicked

Figure A.8: Organic Compounds



Personal Hotspot : 2 connection(s),Used 479 MB

MPTS

Oxygen

H2O:Water

CO2:Carbon Dioxide

SiO2:Silica

Al2O3:Aluminium Oxide

Fe2O3:Iron Oxide

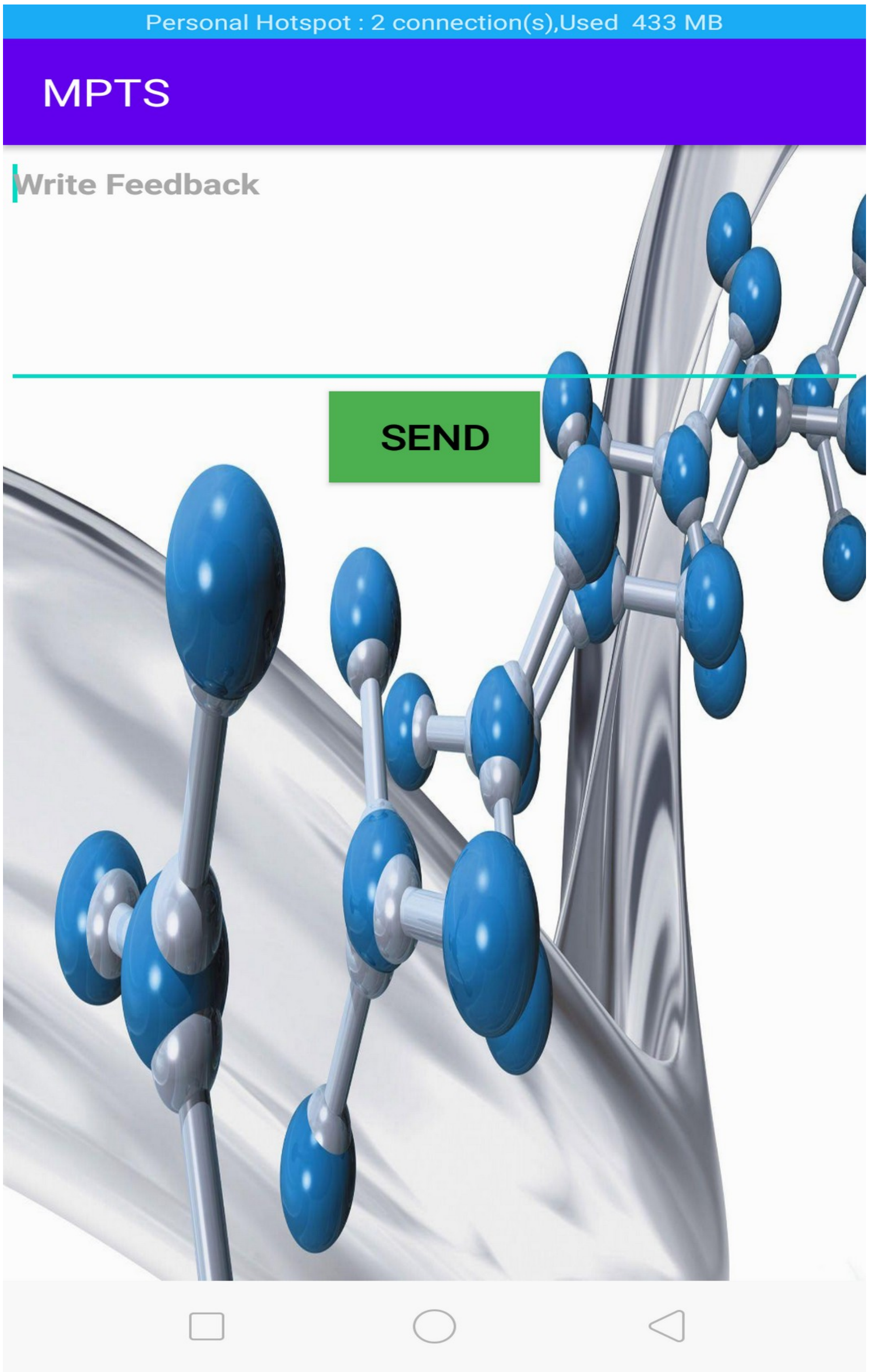
H2O2:Hydrogen Peroxide

KO2:Potassium Superoxide

Na4SiO4,Na2SiO3 and Na2Si2O5

text is Clicked

Figure A.9: Inorganic compounds



Personal Hotspot : 2 connection(s),Used 433 MB

MPTS

Write Feedback

Figure A.10: Feedback Activity

## A.2 SAMPLE CODE

**RegisterActivity.java**

```
package com.example.mpts;



        import androidx.annotation.NonNull;

        import androidx.appcompat.app.AppCompatActivity;

        import android.content.Intent;

        import android.os.Bundle;

        import android.util.Log;

        import android.view.View;

        import android.widget.Button;

        import android.widget.EditText;


        import android.widget.TextView;

        import android.widget.Toast;

        import com.google.android.gms.tasks.OnCompleteListener;

        import com.google.android.gms.tasks.OnFailureListener;

        import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;

import com.google.firebase.FirebaseException;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.PhoneAuthCredential;

import com.google.firebase.auth.PhoneAuthProvider;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.rilixtech.widget.countrycodepicker.CountryCodePicker;

import java.util.concurrent.TimeUnit;

public class RegisterActivity extends AppCompatActivity {

    FirebaseAuth fAuth;

    String phoneNumber = "+911922555777";

    String otpCode = "192256";

    String verificationId;

    EditText phone, optEnter;

    Button next;

    CountryCodePicker countryCodePicker;

    PhoneAuthCredential credential;
```

```
Boolean verificationOnProgress = false;

ProgressBar progressBar;

TextView state, resend;

PhoneAuthProvider.ForceResendingToken token;

FirebaseFirestore fStore;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_register);

    phone = findViewById(R.id.phone);

    optEnter = findViewById(R.id.codeEnter);

    countryCodePicker = findViewById(R.id.ccp);

    next = findViewById(R.id.nextBtn);

    fAuth = FirebaseAuth.getInstance();

    fStore = FirebaseFirestore.getInstance();

    progressBar = findViewById(R.id.progressBar);

    state = findViewById(R.id.state);

    resend = findViewById(R.id.resendOtpBtn);

    resend.setOnClickListener(new View.OnClickListener() {

        @Override
```

```
            public void onClick(View v) {

            }

        });

        next.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                if (!phone.getText().toString().isEmpty() && phone.getText().toString().length()
== 10) {

                    if (!verificationOnProgress) {

                        next.setEnabled(false);

                        progressBar.setVisibility(View.VISIBLE);

                        state.setVisibility(View.VISIBLE);

                        String phoneNum = "+" + countryCodePicker.getSelectedCountryCode() +
phone.getText().toString();

                        Log.d("phone", "Phone No.: " + phoneNum);

                        requestPhoneAuth(phoneNum);

                    } else {

                        next.setEnabled(false);

                        optEnter.setVisibility(View.GONE);

                        progressBar.setVisibility(View.VISIBLE);

                        state.setText("Logging in");
                        state.setVisibility(View.VISIBLE);
                        otpCode = optEnter.getText().toString();
                        if (otpCode.isEmpty()) {
                            optEnter.setError("Required");
                            return;
                        }
                        credential = PhoneAuthProvider.getCredential(verificationId, otpCode);
                        verifyAuth(credential);
                    }
                } else {
                    phone.setError("Valid Phone Required");
                }
            }
        });
    }
    private void requestPhoneAuth(String phoneNumber) {
        PhoneAuthProvider.getInstance().verifyPhoneNumber(phoneNumber, 60L,
TimeUnit.SECONDS, this,
                new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {
                    @Override
                    public void onCodeAutoRetrievalTimeOut(String s) {
```

```
            super.onCodeAutoRetrievalTimeOut(s);
            Toast.makeText(RegisterActivity.this, "OTP Timeout, Please Re-generate
the OTP Again.", Toast.LENGTH_SHORT).show();
            resend.setVisibility(View.VISIBLE);
        }
        @Override
        public void onCodeSent(String s, PhoneAuthProvider.ForceResendingToken
forceResendingToken) {
            super.onCodeSent(s, forceResendingToken);
            verificationId = s;
            token = forceResendingToken;
            verificationOnProgress = true;
            progressBar.setVisibility(View.GONE);
            state.setVisibility(View.GONE);
            next.setText("Verify");
            next.setEnabled(true);
            optEnter.setVisibility(View.VISIBLE);
        }
        @Override
        public void onVerificationCompleted(PhoneAuthCredential
phoneAuthCredential) {
            verifyAuth(phoneAuthCredential);
```

```
          }

          @Override

          public void onVerificationFailed(FirebaseException e) {

            Toast.makeText(RegisterActivity.this, e.getMessage(),
Toast.LENGTH_SHORT).show();

          }

        });

  }

  private void verifyAuth(PhoneAuthCredential credential) {

    fAuth.signInWithCredential(credential).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {

      @Override

      public void onComplete(@NonNull Task<AuthResult> task) {

        if (task.isSuccessful()) {

          Toast.makeText(RegisterActivity.this, "Phone Verified." +
fAuth.getCurrentUser().getUid(), Toast.LENGTH_SHORT).show();

          checkUserProfile();

        } else {

          progressBar.setVisibility(View.GONE);

          state.setVisibility(View.GONE);

          Toast.makeText(RegisterActivity.this, "Can not Verify phone and Create
Account.", Toast.LENGTH_SHORT).show();
```

```
                }
            }
        });
    }
    @Override
    protected void onStart() {
        super.onStart();
        if (fAuth.getCurrentUser() != null) {
            progressBar.setVisibility(View.VISIBLE);
            state.setText("Checking..");
            state.setVisibility(View.VISIBLE);
            checkUserProfile();
        }
    }
    private void checkUserProfile() {
        DocumentReference docRef =
fStore.collection("users").document(fAuth.getCurrentUser().getUid());
        docRef.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>()
{
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                if (documentSnapshot.exists()) {

                    startActivity(new Intent(getApplicationContext(), MainActivity.class));

                    finish();

                } else {

                    //Toast.makeText(Register.this, "Profile Do not Exists.",
Toast.LENGTH_SHORT).show();

                    startActivity(new Intent(getApplicationContext(), DetailsActivity.class));

                    finish();

                }

            }

        }).addOnFailureListener(new OnFailureListener() {

            @Override

            public void onFailure(@NonNull Exception e) {

                Toast.makeText(RegisterActivity.this, "Profile do not exists",
Toast.LENGTH_SHORT).show();

            }

        });

}

}
```

**HydrogenActivity.java**

```
package com.example.mpts;


import androidx.appcompat.app.AppCompatActivity;


import android.os.Bundle;


public class HydrogenActivity extends AppCompatActivity {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hydrogen);
    }
}
```

**RegisterActivity.java**

```
package com.example.mpts;
```

```
import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;


import android.os.Bundle;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.TextView;


import com.google.firebase.database.ChildEventListener;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;


import java.util.ArrayList;


public class InorganicActivity extends AppCompatActivity {


    DatabaseReference reffff = FirebaseDatabase.getInstance().getReference();




    ArrayList<String> inor = new ArrayList<>();
    ListView qList;



    TextView mInorganicTv;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inorganic);


        mInorganicTv = findViewById(R.id.title11);
        mInorganicTv.setText(getIntent().getExtras().getString("element"));
        String el = mInorganicTv.getText().toString();
        String ele = el + "inorg";
        DatabaseReference fRef = reffff.child(ele);
        qList = findViewById(R.id.listView2);
        final ArrayAdapter arrayAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1,inor);
        qList.setAdapter(arrayAdapter);
        fRef.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {
                String value = dataSnapshot.getValue(String.class);
                inor.add(value);
                arrayAdapter.notifyDataSetChanged();
            }

            @Override
            public void onChildChanged(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {

            }

            @Override
            public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {

            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
```

```
    }
}
```

**FeedbackActivity.java**

```
package com.example.mpts;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class FeedbackActivity extends AppCompatActivity {

    EditText eText;
    Button b1;
    Feed feedBack;
    DatabaseReference ref =
FirebaseDatabase.getInstance().getReference().child("Feedback");


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_feedback);


        eText = (EditText)findViewById(R.id.ed1);
        b1 = (Button)findViewById(R.id.bt1);
        feedBack = new Feed();


        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String f = eText.getText().toString();
                eText.setVisibility(View.GONE);
                feedBack.setFd(f);
                ref.push().setValue(feedBack);
                Toast.makeText(getApplicationContext(), "Your Feedback is
Successfully Submitted", Toast.LENGTH_SHORT).show();


            }
        });
    }
```

```
}
```

**AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mpts">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".ZoomActivity"></activity>
        <activity android:name=".InorganicActivity" />
        <activity android:name=".CompundsActivity" />
        <activity android:name=".FeedbackActivity" />
        <activity android:name=".AboutActivity" />
        <activity android:name=".ChatActivity" />
        <activity android:name=".UsesActivity" />
        <activity android:name=".FeatureActivity" />
        <activity android:name=".HydrogenActivity" />
        <activity
            android:name=".AnotherActivity"
            android:parentActivityName=".ListActivity" />
        <activity android:name=".ListActivity" />
        <activity android:name=".DetailsActivity" />
        <activity
            android:name=".RegisterActivity"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".MainActivity"
            android:theme="@style/AppTheme.NoActionBar" />
    </application>
```

```
</manifest>
```

**CompoundActivity.java**

```
package com.example.mpts;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.ArrayList;

public class CompundsActivity extends AppCompatActivity {
    DatabaseReference refff = FirebaseDatabase.getInstance().getReference();



    ArrayList<String> comp = new ArrayList<>();
    ListView tList;



    TextView mCompoundTv;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_compunds);



        mCompoundTv = findViewById(R.id.title10);
        mCompoundTv.setText(getIntent().getExtras().getString("element"));
        String el = mCompoundTv.getText().toString();
        String ele = el + "comp";
        DatabaseReference fRef = refff.child(ele);
        tList = findViewById(R.id.listView1);
        final ArrayAdapter arrayAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1,comp);
```

```
        tList.setAdapter(arrayAdapter);
        fRef.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {
                String value = dataSnapshot.getValue(String.class);
                comp.add(value);
                arrayAdapter.notifyDataSetChanged();


            }


            @Override
            public void onChildChanged(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {


            }


            @Override
            public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {


            }


            @Override
            public void onChildMoved(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }
}
```

**FeatureActivity.java**

```
package com.example.mpts;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.sql.Ref;
import java.util.ArrayList;

public class FeatureActivity extends AppCompatActivity {

  DatabaseReference ref = FirebaseDatabase.getInstance().getReference();



  ArrayList<String> fea = new ArrayList<>();
  ListView fList;



   TextView mFeatureTv;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_feature);


        mFeatureTv = findViewById(R.id.title3);
        mFeatureTv.setText(getIntent().getExtras().getString("element"));
        String el = mFeatureTv.getText().toString();
        DatabaseReference fRef = ref.child(el);
        fList = findViewById(R.id.listView);
        final ArrayAdapter arrayAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1,fea);
        fList.setAdapter(arrayAdapter);
        fRef.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String
s) {
                String value =dataSnapshot.getValue(String.class);
                fea.add(value);
                arrayAdapter.notifyDataSetChanged();


            }


            @Override
            public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {
            }

            @Override
            public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }
}
```

**DetailsActivity.java**

```
package com.example.mpts;
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashMap;
import java.util.Map;

public class DetailsActivity extends AppCompatActivity {
```

```
public static final String TAG = "TAG";
Button saveBtn;
FirebaseAuth fAuth;
FirebaseFirestore fStore;
EditText firstName, lastName, email;
String userID;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_details);

    firstName = findViewById(R.id.firstName);
    lastName = findViewById(R.id.lastName);
    email = findViewById(R.id.emailAddress);
    saveBtn = findViewById(R.id.saveBtn);

    fAuth = FirebaseAuth.getInstance();
    fStore = FirebaseFirestore.getInstance();
    userID = fAuth.getCurrentUser().getUid();

    saveBtn.setOnClickListener(new View.OnClickListener() {
        @Override
```

```
      public void onClick(View v) {
        if (firstName.getText().toString().isEmpty() ||
lastName.getText().toString().isEmpty() || email.getText().toString().isEmpty()) {
          Toast.makeText(DetailsActivity.this, "Fill the required Details",
Toast.LENGTH_SHORT).show();
          return;
        }



        DocumentReference docRef = fStore.collection("users").document(userID);

        Map<String, Object> user = new HashMap<>();
        user.put("first", firstName.getText().toString());
        user.put("last", lastName.getText().toString());
        user.put("email", email.getText().toString());

        docRef.set(user).addOnSuccessListener(new OnSuccessListener<Void>() {
          @Override
          public void onSuccess(Void aVoid) {
            Log.d(TAG, "onSuccess: Use.r Profile Created" + userID);
            startActivity(new Intent(getApplicationContext(), MainActivity.class));
            finish();
          }
        }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Log.d(TAG, "onFailure: Failed to Create User " + e.toString());
                }
            });
        }
    });
  }
}
```

**CustomeFilter.java**

```
package com.example.mpts;

import android.widget.Filter;

import java.util.ArrayList;

public class CustomFilter extends Filter {

    ArrayList<Model> filterList;
    MyAdapter adapter;



    public CustomFilter(ArrayList<Model> filterList, MyAdapter adapter) {
        this.filterList = filterList;
        this.adapter = adapter;
    }

    @Override
    protected FilterResults performFiltering(CharSequence constraint) {

        FilterResults results = new FilterResults();

        if (constraint != null && constraint.length() > 0) {

            constraint = constraint.toString().toUpperCase();

            ArrayList<Model> filterModels = new ArrayList<>();

            for (int i = 0; i < filterList.size(); i++) {
                if (filterList.get(i).getTitle().toUpperCase().contains(constraint)) {

                    filterModels.add(filterList.get(i));
                }
            }

            results.count = filterModels.size();
            results.values = filterModels;
        }


        else {
            results.count = filterList.size();
            results.values = filterList;
        }


        return results;
    }


    @Override
    protected void publishResults(CharSequence constraint, FilterResults results)
{


        adapter.models = (ArrayList<Model>) results.values;
        adapter.notifyDataSetChanged();


    }
}
```

**MyAdapter.java**

```
package com.example.mpts;


import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Filter;
import android.widget.Filterable;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.io.ByteArrayOutputStream;
import java.util.ArrayList;

public class MyAdapter extends RecyclerView.Adapter<MyHolder> implements Filterable {

    Context c;
    ArrayList<Model> models, filterList;
    CustomFilter filter;



    public MyAdapter(Context c, ArrayList<Model> models) {
```

```
    this.c = c;

    this.models = models;

    this.filterList = models;

}


@NonNull

@Override

public MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {


    View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.row, null);


    return new MyHolder(view);

}


@Override

public void onBindViewHolder(@NonNull final MyHolder myHolder, int i) {


   myHolder.mTitle.setText(models.get(i).getTitle());

   myHolder.mDesc.setText(models.get(i).getDescription());

   myHolder.mImageView.setImageResource(models.get(i).getImg());


    /*one activity*/

   myHolder.setClickElement(new ClickElement() {
        @Override
        public void onItemClickListener(View v, int position) {


            String gTitle = models.get(position).getTitle();

            String gDesc = models.get(position).getDescription();

            BitmapDrawable bitmapDrawable =
(BitmapDrawable)myHolder.mImageView.getDrawable();


            Bitmap bitmap = bitmapDrawable.getBitmap();


            ByteArrayOutputStream stream = new ByteArrayOutputStream();


            bitmap.compress(Bitmap.CompressFormat.PNG,100, stream);


            byte[] bytes = stream.toByteArray();




            Intent intent = new Intent(c, AnotherActivity.class);

            intent.putExtra("iTitle", gTitle);

            intent.putExtra("iDesc", gDesc);

            intent.putExtra("iImage", bytes);

            c.startActivity(intent);
```

```
    }
});

/*use different activities*/
/*myHolder.setClickElement(new ClickElement() {
    @Override
    public void onItemClickListener(View v, int position) {
        if (models.get(position).getTitle().equals("Hydrogen")){
        }
        if (models.get(position).getTitle().equals("Helium")){
        }
        if (models.get(position).getTitle().equals("Lithium")){
        }
        if (models.get(position).getTitle().equals("Beryllium")){
        }
        if (models.get(position).getTitle().equals("Boron")){
        }
        if (models.get(position).getTitle().equals("Carbon")){
        }
        if (models.get(position).getTitle().equals("Nitrogen")){
        }
        if (models.get(position).getTitle().equals("Oxygen")){
        }
```

```
                if (models.get(position).getTitle().equals("Fluorine")){
                }
                if (models.get(position).getTitle().equals("Neon")){
                }
            }
        });*/

    }

@Override
    public int getItemCount() {
        return models.size();
    }

    @Override
    public Filter getFilter() {

        if (filter == null) {
            filter = new CustomFilter(filterList, this);
        }

        return filter;
    }}
```