**REPORT**

**ON**

**SIMPLE MACHINE LEARNING**

**IRIS DATASET**

****

**ACKNOWLEDGEMENT**

Apart from efforts, the success of this project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I take immense pleasure in thanking and warmly acknowledging the continuous encouragement, invaluable supervision, suggestions and inspired guidance offered by my mentor.Dr.Jason Brownlee , professional developer and a machine learning practitioner ,Tutor at http://machinelearningmastery.com, in bringing this report to a successful completion.

I also express my sincere thanks to all my friends who have patiently extended all sorts of help for accomplishing this undertaking.

Finally we extend our gratefulness to one and all that are directly or indirectly involved in the successful completion of this project work.

**Priyanka Agrawal**

**KIIT UNIVERSITY**

**ABSTRACT**

Iris dataset is a classic dataset, it is one of the modern examples of stastical classification.The dataset is a collection of measurements of several iris flowers.These measurements helps us to distinguish different species of the flowers.Today , species can be easily identified by using DNA fingerprints , but time when DNA’s role in genetics had not yet been discovered.Species can be identified by using supervised algorithm on the dataset.

It is a classification problem , identifying to which a set of categories (species for this dataset) a new observation belongs to,on the basis of training dataset containing observations(by prediction models) category (species) can be known.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.NO |  | CONTENTS | PAGE NO. |
| 1. |  | Introduction | 5 |
|  | 1.1 | Methodology | 5 |
|  | 1.2 | Risks Involved | 6 |
|  | 1.3 | Purpose | 8 |
|  | 1.4 | Scope | 9 |
|  | 1.5 | Tools Used | 10 |
| 2. |  | Objective | 11 |
| 3. |  | Project Planning | 12 |
|  | 3.1 | Requirement Gathering | 12 |
|  | 3.2 | Functional Requirements | 13 |
|  | 3.3 | Non-Functional Requirements | 15 |
|  | 3.4 | Constraints | 16 |
| 4. |  | Design | 17 |
|  | 4.1 | Use Case Diagram | 17 |
|  | 4.2 | Class Diagram | 19 |
|  | 4.3 | Sequence Diagram | 21 |
| 5. |  | Implementation | 26 |
| 6. |  | Conclusion | 27 |
| 7. |  | Bibliography | 28 |

**1. DEFINE PROBLEM**

This dataset consists of four attributes using which we can create classification between different species of iris flowers.

The following four attributes of each plant was measured.

* sepal length
* sepal width
* petal length
* petal width

This dataset has four features. Additionally, for each plant, the species was recorded. The problem we want to solve is, "Given these examples, if we see a new flower out in the field, could we make a good prediction about its species from its measurements?"

For the optimal result we have to evaluate different algorithms on the dataset and compare the accuracy of each model or algorithm .

Choose the most accurate model.Our aim here will also be to get the accuracy within our expected margin ,suggesting we may have an accurate and a reliably accurate model.

By choosing the most accurate model we can easily classify the flowers by having a look into their measurement attributes.The more accurate the model will be less error will be occuring while the classification of the species.

**1.2 STEPS INVOLVED**

1) **Prepare Data**

* The first basic step is to collect or acquire the dataset.
* Load the Iris dataset the easy way.
* There are different methods available for loading the datasets in R.

Ex-read.csv,read\_csv,etc.

* Use method that gives best way to represent data in R.

Ex-Tribble,Data Frame,etc.

* Separate the data into a training dataset and a validation dataset

2) **Evaluate Algorithms**

* We are considering two algorithms here for predicting the species of flower and choose the best model.
* k-Nearest Neighbors (kNN).
* Random Forest (RF).

3) **Improve and Present Results**

* Compare the results of different models.
* Get a summary of what was used to train the model and accuracy achieved.

MAKE PREDICTIONS

This is the final step which we need to perform after selecting the model ,Comparing the results from different models we have to select the best model of it and carry on further process with it.

Now we want to get an idea of the accuracy of the model on our validation set.

This will give us an independent final check on the accuracy of the best model. It is valuable to keep a validation set just in case you made a slip during such as overfitting to the training set or a data leak. Both will result in an overly optimistic result.

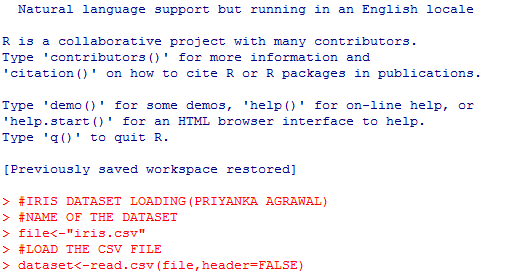
We can run the model directly on the validation set and summarize the results .

**LOAD THE DATA**

We are going to use Iris flower dataset.

All the observed flowers here belong to three species.

**Load from CSV file**



You we have the iris data loaded in R and accessible through the dataset variable.

**Create Validation Dataset**

We need to know that the model we created is any good.

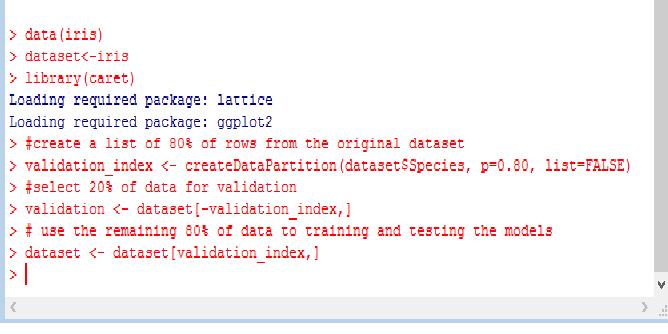
Later, we will use methods to estimate the accuracy of the models that we create on unseen data. We also want a estimate of the accuracy of the best model on unseen data by evaluating it on actual unseen data.  
That is, we are going to hold back some data that the algorithms will not get to see and we will use this data to get a second and independent idea of how accurate the best model might actually be.

We will split the loaded dataset into two, 80% of which we will use to train our models and 20% that we will hold back as a validation dataset.

By using the 80% of data that is training data will be used for the testing of the models.The remaining is used for validation after choosing the model.

Now we have to go for the process of Data Partioning .

For Machine Learning in R we use the Package “caret”,it provides us different functionalities , createDataPartition is used here.



**SUMMARIZE DATASET**

After collecting the data and partioning it there is need to have a look on the dataset’s arguments,number of records,Class of datatype,etc.

Summary of dataset helps us to get proper insight of the data and it’s structure.By using the functions provided by R we can easily see the following aspects of a dataset.

 Dimensions of the dataset.

 Types of the attributes.

 Peek at the data itself.

 Levels of the class attribute.

 Breakdown of the instances in each class.

 Statistical summary of all attributes

Dimensions of dataset

We can get a quick idea of how many instances (rows) and how many attributes (columns) the data contains with the dim function.

Types of attributes

It is a good idea to get an idea of the types of the attributes. They could be doubles, integers, strings, factors and other types.

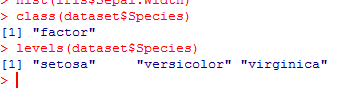
Knowing the types is important as it will give us an idea of how to better summarize the data we have and the types of transforms we might need to make to data before it can be used for model .

Peek at the data

It is also always a good idea to actually have a look of the data.For this we can use head() or tail() ,by default it shows 6 records.Among thousands of records it helps to show the information of few of the records.

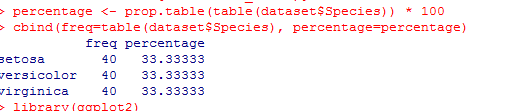
**Levels of the Class**

There are different types of variables available in the R environment.To categorize among them we need to find the class of the variables.The variables available in R are Numeric,Vector,Factors,etc. If we know the type of variables then it would be much easier to work on datasets.



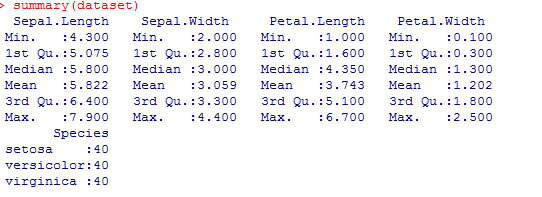
**Class Distribution**

Now we have to see the distribution of records or data according to a particular variable in our training dataset.The distribution should be almost equal to give us a good result, otherwise in other case the result will be much dependent on the variable having higher distribution in the dataset.Ex-In our dataset the distribution among the three Species should be equal.



**Stastical Summary**

We can now have a look at the summary of each attribute.This includes the mean , maximum , minimum incluing some percentiles.



We can see that all of the numerical values have the same scale (centimeters) and similar ranges [0,8] centimeters.

**Visualize Dataset**

Now we have a basic idea about the data , structure of data , types of data.

We can increase our understanding of data by having visualization of dataset.

For this dataset we are going to use two different types of plots for visualization

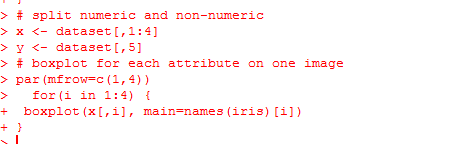
1. Univariate plots to better understand each attribute.
2. Multivariate plots to better understand the relationships among the attributes.

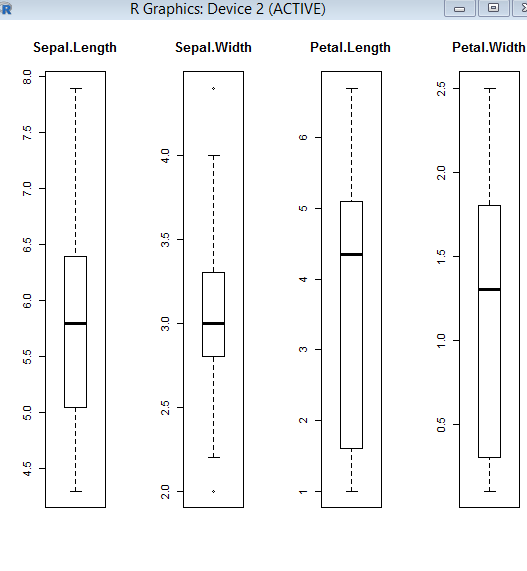
**Univariate Plots**

These are the plots of individual variables.

It helps us to understand the behaviour of individual attribute in a dataset.

Ex-we can the max,min ,mean values of the attribute using **boxplot( ).**

****

****

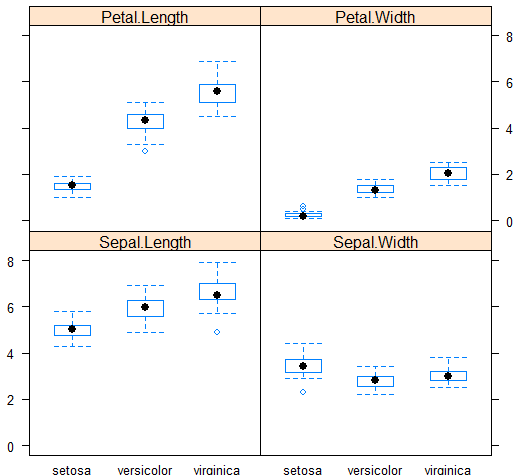
**HISTOGRAM**

We can also create a barplot of the Species class variable to get a graphical representation of the class distribution (In this case as we have testing data partition according to Species variable so the plot should be equal for them).



**MULTIVARIATE PLOTS**

Now we can have a look at the plots of each input variable,broken into plots of each class.This will help us to see the separation between the different species of flowers.

****

**EVALUATE ALGORITHMS**

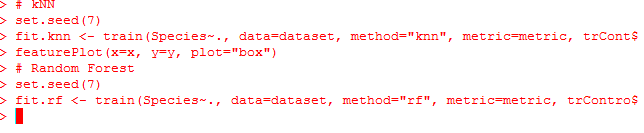
Now its time to evaluate some of the basic algorithms of classification models and estimate their accuracy on the unseen data.

Let’s evaluate the two widely accepted algorithms

1.-K-Nearest Neighbours(KNN)

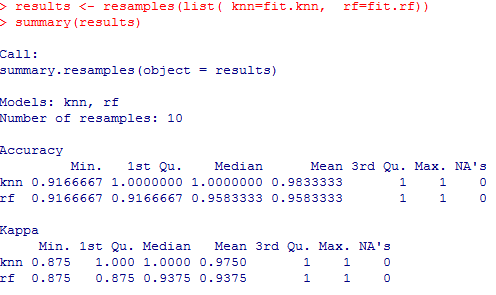
2.-Random Forest(RF)

The functionality of these two algorithms are already provided in the R-Environment, in the CARET(Classification and Regression Training) package.

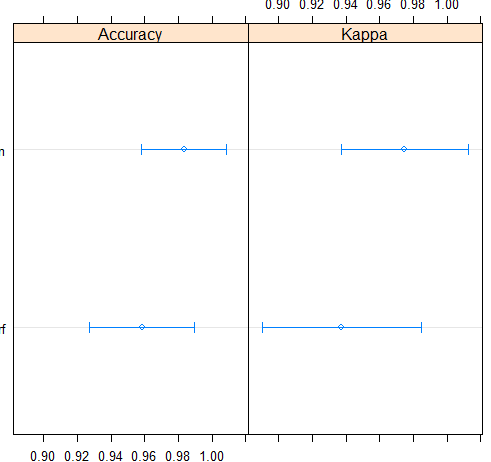


**SELECT BEST MODEL**

We create report on accuracy of each model by combining their results in a list and evaluating a summary function.

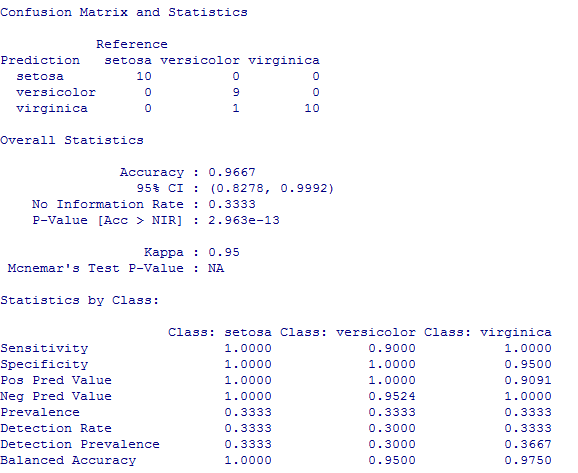


By looking at the below plot of results we can see the accuracy of both the models and select one of them as our final model.



**PREDICTIONS**

Now we will use the model for prediction of flower species on the validation dataset,which we have separated earlier from the training dataset.The result what we get here is almost correct

****

**R3. Update Notice**

**Description**: **Updates any important notices related to the student.**

**Input:** Roll\_no

**Output**: Update Notice

**Processing**: Add contact to Database, Delete contact to Database, Add new update flag to Database.

**R3.1 Update Timetable and attendence**

**Description**: **Updates time table and attendance according to the branch.**

**Input:** Branch

.**Output**: Time Table

**Processing**: updates to DB, Add new update flag to DB

**R3.3 View notices And Time table**

**Description**: **Student can view time table and notices.**

**Input:** Student login id

**Output**: important notices and placement related info.

**Processing**: Updates placement details, notices according to the branch (inferred from the roll no)

**R4. Notification**

**Description**: **Students get notification on class timings and new notice updates**

**Input:** Login details

**Output**: Notifications on phone.

**Processing**: Add notice to Database, Add new update flag to Database which is connected to the system and is notified to the user.

**Non-Functional Requirements:**

**24x7 Availability**: The smart traffic system is available 24X7 managing traffic all the time.

**Performance Requirements**: Performance of the system depends on the response time and the speed of the data submission. The response time of the system is direct and the application is real-time. System should have a fast response time which depends on the efficiency of the implemented algorithm.If the system is not real-time it is not useful.

The first version of the system will have limited implementation hence there will be no need for a large network. However, it may grow depending on the increase in usage.

**Safety Requirements**: System has to check-

* If Database content is syntactically well formed.
* If another server is not running on the same port which might not allow Apache-MySQL server to run.
* If web forms with the services processing form input are consistent.
* Statically safe binding of code of session operations to variables defined with session scope.

**Security Requirements**: For security of the system the technique known as database replication should be used so that all important data is kept safe. In case of crash, the system should be able to backup and recover the data. Traffic Manager should only be given the access for database monitoring and insights from the data.

**Software Quality Attribute**: The system will have a simple and we will be using basic sensors which can easily collect dataset and act upon it smartly. Flexible service based architecture will be highly desirable for future extension.

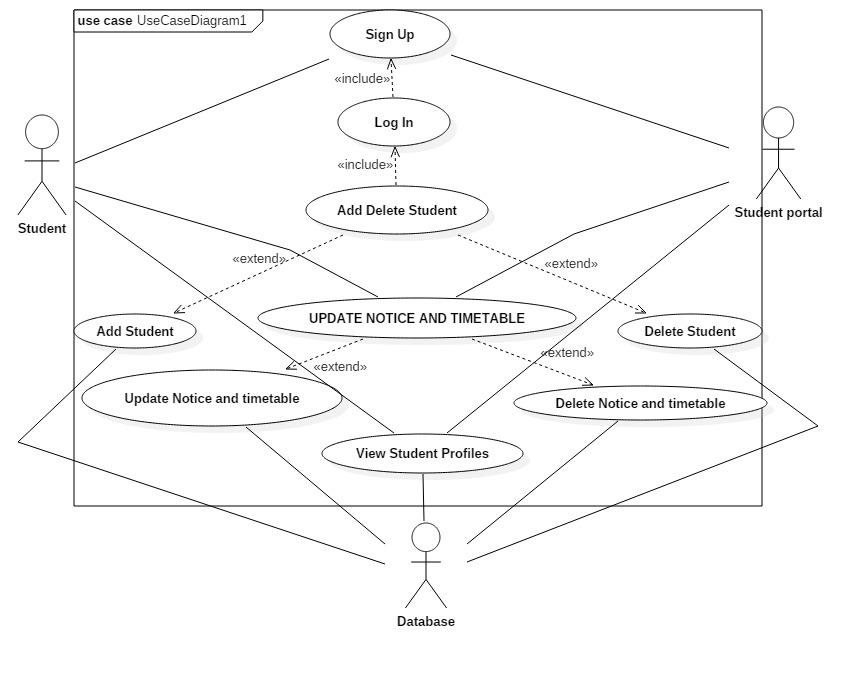
**3.4 Constraints:**

* The application will be useful only in the high or variable traffic areas.
* The citizen have to lookup the webpage for the parking space availibility from a table format.
* Only citizen having smartphone can utilize the services provided.
* Citizens who have the knowledge of accessing net facilities can only utilize the offered services

4. DESIGN

**4.1Use-Case Diagram**

An **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.



Use Case Diagram for Traffic Management

Use Case Report:

|  |  |
| --- | --- |
| Use Case | Description |
| Signup | Register studname and password with server for use of service |
| Login | Log in to students’s personal profile to access data |
| Add contact | Adds new contact details to profile |
| Add Contact Details | Adds new contact details to profile |
| Delete Contact Details | Deletes contact detail on profile |
| Add/Delete Profile | Adds new contact details to profile |
| Add Contact | Adds new personal friend to Friend List |
| Delete Contact | Deletes personal friend from Friend List |
| View Profile | View profile of a friend or colleague |

**4.2Class Diagram**

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), a **class diagram** in the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) is a type of static structure diagram that describes the structure of a system by showing the system's [classes](https://en.wikipedia.org/wiki/Class_(computer_science)), their attributes, operations (or methods), and the relationships among objects.

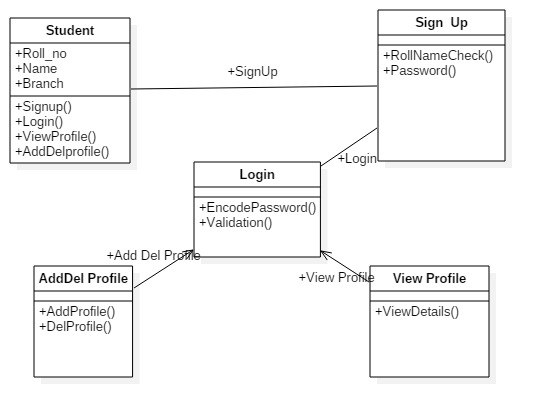
The class diagram is the main building block of [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) modelling. It is used both for general [conceptual modelling](https://en.wikipedia.org/wiki/Conceptual_model) of the systematics of the application, and for detailed modelling translating the models into [programming code](https://en.wikipedia.org/wiki/Programming_code). Class diagrams can also be used for [data modeling](https://en.wikipedia.org/wiki/Data_modeling). The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

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

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

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a number of subclasses.

In order to further describe the behaviour of systems, these class diagrams can be complemented by a [state diagram](https://en.wikipedia.org/wiki/State_diagram) or [UML state machine](https://en.wikipedia.org/wiki/UML_state_machine).



Class Diagram of Student Portal

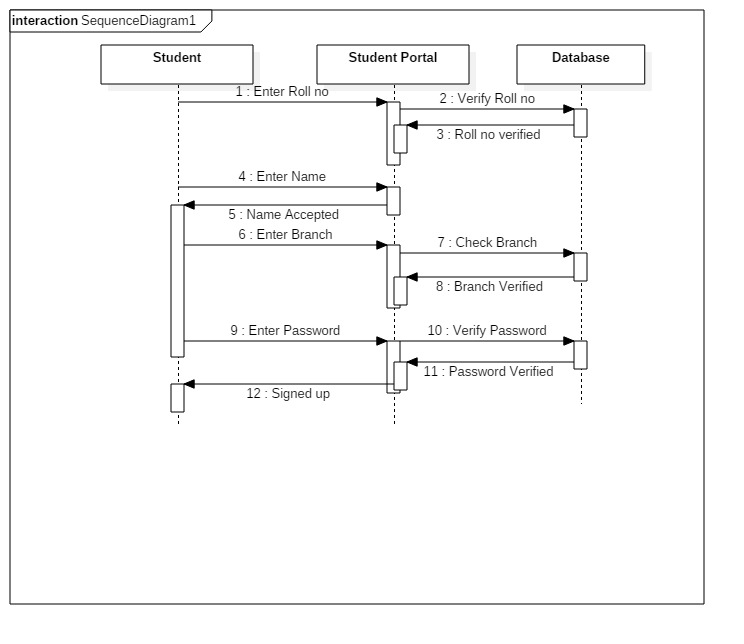
**4.3 Sequence diagram**

A **Sequence diagram** is an [interaction diagram](https://en.wikipedia.org/wiki/Interaction_diagram) that shows how objects operate with one another and in what order. It is a construct of a [message sequence chart](https://en.wikipedia.org/wiki/Message_sequence_chart).

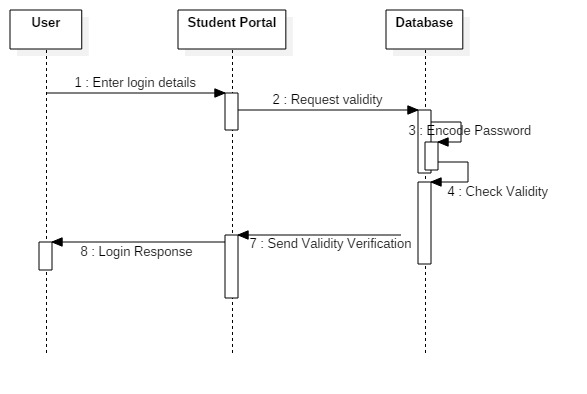
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case 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

**1.Sign Up:**

****

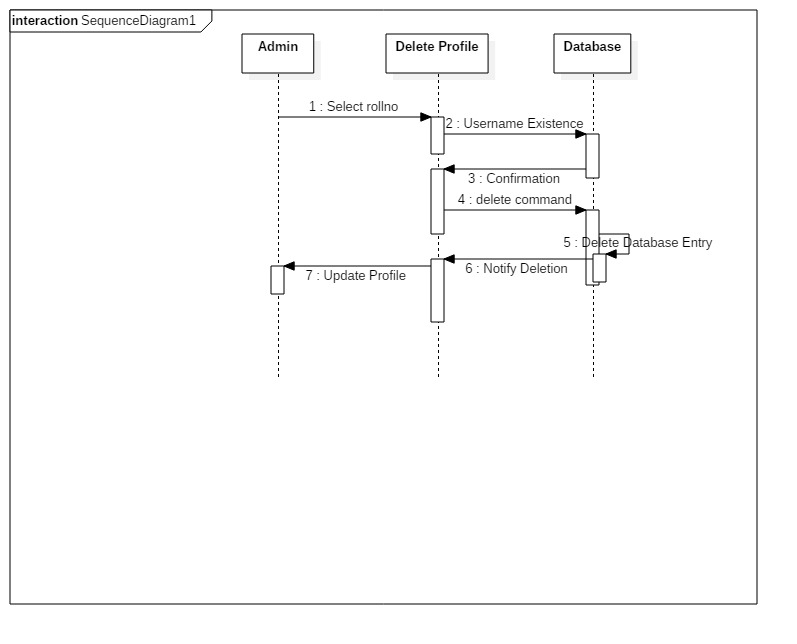
**2.Login**

****

Sequence Diagram for Login

1. The User enters login details.
2. The contact application requests validity to the database
3. The database checks the validity.
4. The database sends result
5. The contact app. gives the logical response to the user

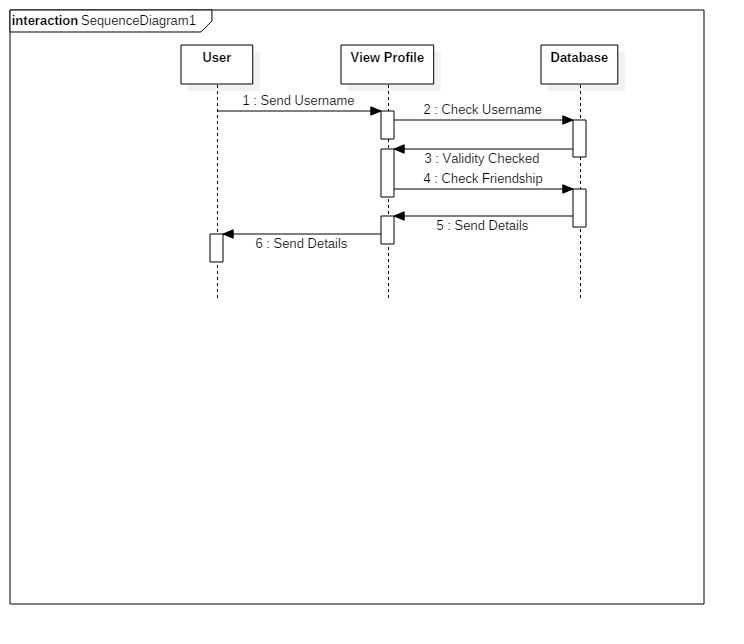
**3. Delete Profile**



Sequence Diagram for Delete Profile

1. The User 1 sends username.
2. The contact application checks for username existence.
3. The database checks the existence.
4. The database sends confirmation.
5. The contact app. checks if already friends.
6. The database deletes database entry.
7. The database notifies deletion to contact app
8. The contact app updates profile to User1 and User 2.

**4. View Profile**



Sequence Diagram for View Profile

1. The User sends username.
2. The contact application checks for username existence.
3. The database checks the existence.
4. The database sends confirmation.
5. The contact app. checks if already friends
6. The database sends details to contact app.
7. The contact app. sends details to the user.

**IMPLEMENTATION**

-We are going to use PHP and MYSQL to create a database and backend API’s and it will help in storing User’s Profile.

-Using HTML, CSS and java script for the web app and creating the front end of the project to make the software more attractive and User-friendly.

-Java and Xml for Android application so that the android phone users can use the Student Portal application.

**CONCLUSION**

Traffic Management will dynamically access the density of vehicles on each of the lanes in the real-time to handle the congestion in traffic. It will be able to differentiate between lanes according to congestion and provide proirity access to lane manages time with them. It will help in keeping all the information of the lane in a particular area .Because there are many cases where one lane having very few vehicles are provided with same priority which is not efficient.

It will help in dealing with the demerits of the presen. Contact Keeper will take in to consideration all the demerits of today’s world and help in improving the Mobile phone usage in today’s generation.

**BIBLIOGRAPHY**

* https://www.w3schools.com/
* Software Engineering-Rajib Mall
* <https://www.kiitportal.com/>