**SOFTWARE REQUIREMENT SPECIFICATION**

**Technologies used :**

Android Studio 2.3.3

MS Excel

Java Programming for back end development

**Technological Features of different modules :**

1. **Currency Maximizer**

Currency Maximizer involves conversion of an currency into another another one indirectly by performing series of intermediate conversions which leads to maximized final currency. Since conversion rates are volatile and change daily real time conversion rates are taken from a secured site via Internet. For this purpose data is retrieved from a secure Website. Thus real time data is obtained.

After obtaining data, an algorithm for determining the conversions which would give maximum output is added to the module. It is done using

Functions and classes used :

1. get\_usd(); - function used for conversion of Rupees to other currencies
2. get\_eur();. - function used for conversion of Rupees to other currencies
3. get\_gbp(); - function used for conversion of Rupees to other currencies
4. get\_aud(); - function used for conversion of Rupees to other currencies
5. get\_cad(); - function used for conversion of Rupees to other currencies
6. get\_inr(); - function used for conversion of Rupees to other currencies
7. calculate(); - function used maximize currency

Async Jason Task –

AsyncTask enables proper and easy use of the UI thread. This class allows you to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers.

An asynchronous task is defined by a computation that runs on a background thread and whose result is published on the UI thread. An asynchronous task is defined by 3 generic types, called Params, Progress and Result, and 4 steps, called onPreExecute, doInBackground, onProgressUpdate and onPostExecute.

JSON Task object is used to extract data from the website. The execute(URL url) function of this class is used to obtain the source code of the page whose url is passed as a parameter. Once the source code is obtained data is fetched from the buffer.

1. **Stock Analysis**

This module involves analyzing stocks of different companies and plotting them on Graph.

The first page of the module asks the user for company name. Upon selecting the company its current value and respective details are displayed. The user then has options to either analyze the stock through graph or view the location of the Company’s headquarters on Google Map.

The Graphs are plotted depending on time slots specified by the user. 3 time slots are available. Hence, accordingly performance over a week, month or year can be analyzed. Radio buttons are used to specify the take time period input from the user. Upon display of graphs the regression line may also be displayed.

Regression Line (Line of Best Fit) -

A line of best fit is a straight line drawn through the center of a group of data points plotted on a scatter plot. Scatter plots depict the results of gathering data on two variables. The line of best fit shows whether these two variables appear to be correlated and can be used to help identify trends occurring within the dataset. If specified by the user regression line is plotted along with the graph. This gives the user insight of what is happening

Libraries used -

Graph View 4.0.1 –

GraphView is a library for Android to programmatically create  
flexible and nice-looking diagrams.  
It is **easy** to understand, to integrate and to customize.  
Create Line Graphs, Bar Graphs, Point Graphs  
or implement your own custom types.

Line Graph series –

To plot a line diagram in GraphView you use the [LineGraphSeries](http://appsthatmatter.github.io/GraphView/javadoc/com/jjoe64/graphview/series/LineGraphSeries.html" \t "_blank) class.  
By default this will plot a line between the single data points, no matter how much space is between the data points.  
[LineGraphSeries](http://appsthatmatter.github.io/GraphView/javadoc/com/jjoe64/graphview/series/LineGraphSeries.html) has some special options for styling:

* Thickness of the line  
  You can change the thickness of the line via [setThickness](http://appsthatmatter.github.io/GraphView/javadoc/com/jjoe64/graphview/series/LineGraphSeries.html" \l "setThickness-int-" \t "_blank)()
* Background  
  The area below the line can be filled with a color. This feature can be activated via[setDrawBackground](http://appsthatmatter.github.io/GraphView/javadoc/com/jjoe64/graphview/series/LineGraphSeries.html#setDrawBackground-boolean-)(true).  
  You can set the color via setBackgroundColor(). It is recommended to use a semi-transparent color.
* Data Points  
  The single data points can be highlighted with a point in the graph. You activate this via [setDrawDataPoints](http://appsthatmatter.github.io/GraphView/javadoc/com/jjoe64/graphview/series/LineGraphSeries.html" \l "setDrawDataPoints-boolean-" \t "_blank)(true) and change the size of the points via setDataPointsRadius(). The points will always be in the same color than the line itself.
* Custom Paint object  
  If you want to have some more freedom you can change the series to use your own custom paint object for plotting the lines. You could for example generate dotted lines. Set your custom paint via [setCutomPaint](http://appsthatmatter.github.io/GraphView/javadoc/com/jjoe64/graphview/series/LineGraphSeries.html" \l "setCustomPaint-Paint-" \t "_blank)();

1. **Financial Goal Calculator**

Financial Goal aims in calculating approximately savings that a person should to do in order to have the specified amount.

About 7 Goals are provided –

1. Education
2. Buying an asset
3. Residence
4. Marriage
5. Retirement Plan
6. Foreign Trip
7. Other

The user is expected to provide the approximate rate of return and rate of inflation. Upon providing these values the app returns the approximate amount which he user should save and invest each month in order to achieve the goal.

Formulas used:

The module uses inflation rate as well as return rate to compute the final answer. The estimation is based to assumption that inflation rate is constant with time. On basis of these assumptions and basic money grow concepts the needed saving is calculated.

**4. Bank Comparator**

This module involves comparing interest rates on FD and RD provided by different banks.

Initially the interest rates for Fixed Deposits and Recurring Deposits is displayed for each and every bank. Interest rates provided by Banks are time dependent. For eg.

Here interest rates depend on duration of investment. Hence the algorithm has to take into consideration the investment time given by the user, and thereby match it with the slots specified by the banks. Thus, the interest rate is determined and the resulting return is calculated.