OxiMUter

Oxygen Saturation App

Mohammed Balfas
Paul Kaefer
Dan Thomas III
Xioa Zhang

Overview

- Introduction
- Review Functional & Nonfunctional Cases
- Elements of Software Development
- Code & Design Models
- Demonstration

Functional Requirements

Android SDK

OxiMUter will be contructed using Java OOP using Android SDK to work on Android devices

Take Video

OxiMUter will take a short video of the user's finger for analysis.

Output Results

Blood oxygen levels should be displayed on screen after reliable video input

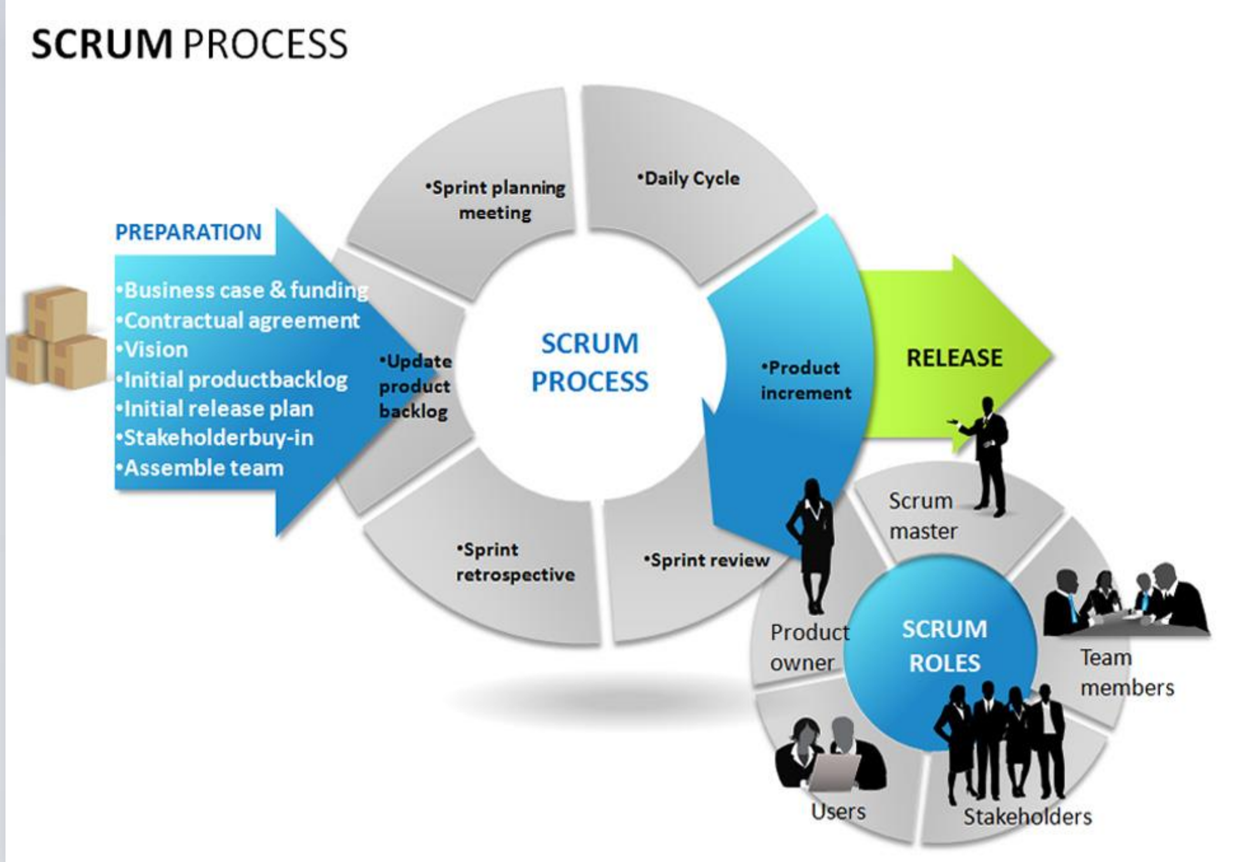
Nonfunctional Requirements

Intuitive UI

The UI should be easy to understand and uncluttered. Menus should assist the user if necessary.

Elements of Software Design

Agile Methodology



Requirement Elicitation

Research and interviews

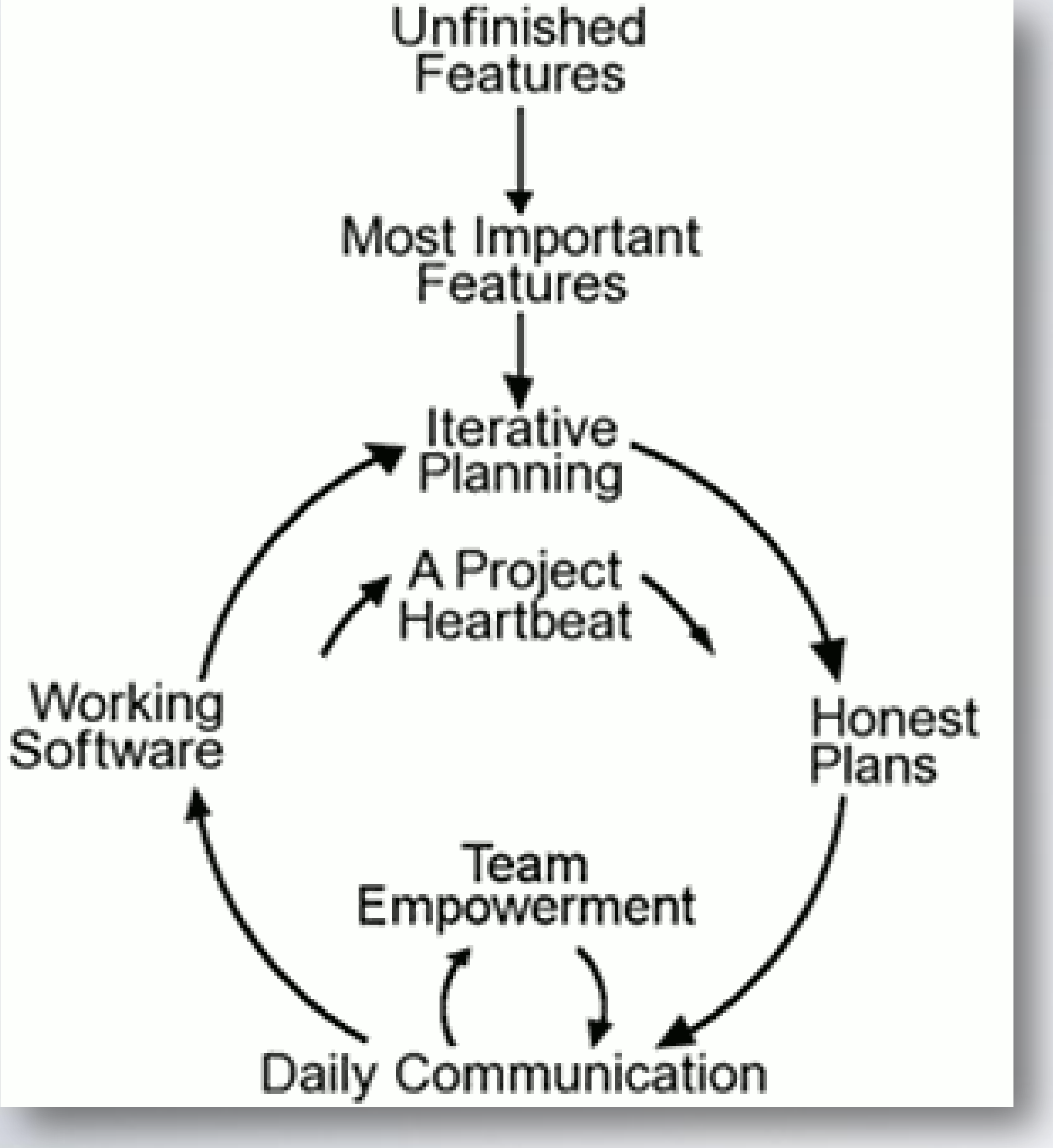
User stories, requirements where gathered immediately to help determine best technological and scientific approach strategies.

Scrum Process

Meetings

Weekly meetings were held and progress tracked.

Agile Methodology



Paired Programming

CreateAndStoreVideo ()

VideoFramesExtraction()

The team was broken down into 2 teams to research reliable components for Android programming

Extreme Programming (XP)

Extensive testing

Software components were first examined, heavily tested, and then implemented in a relatively short time to satisfy use cases

Agile Methodology

Communication

Meeting with Shareholder

Scientific approaches where validated and questions answered

Goals

Weekly Expectations

Weekly teams meetings working towards project deliverables.

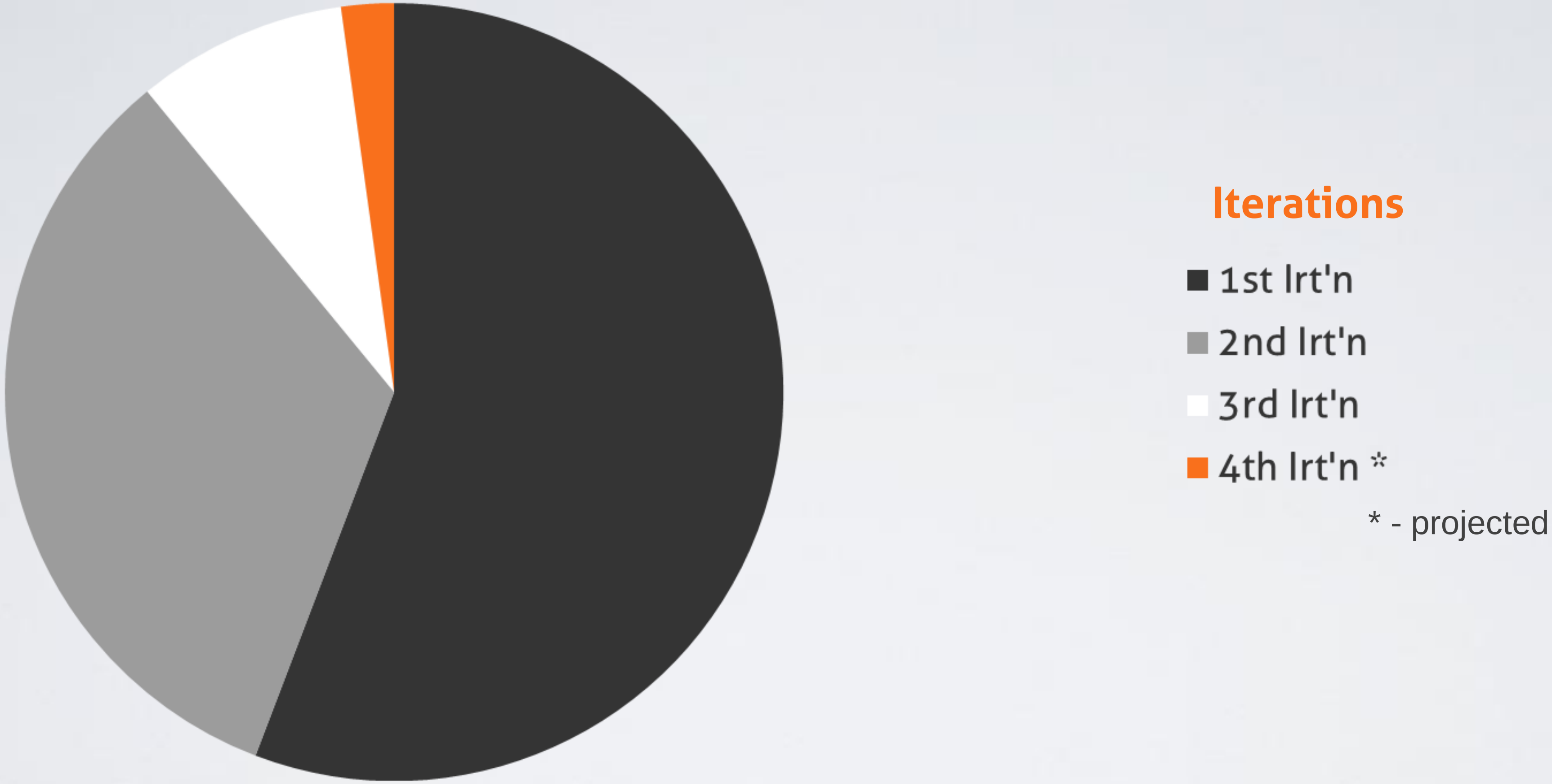
Acceptance Testing

Iteration III

Momentarily, a fairly stable demonstration of a initial prototype of OxiMUter

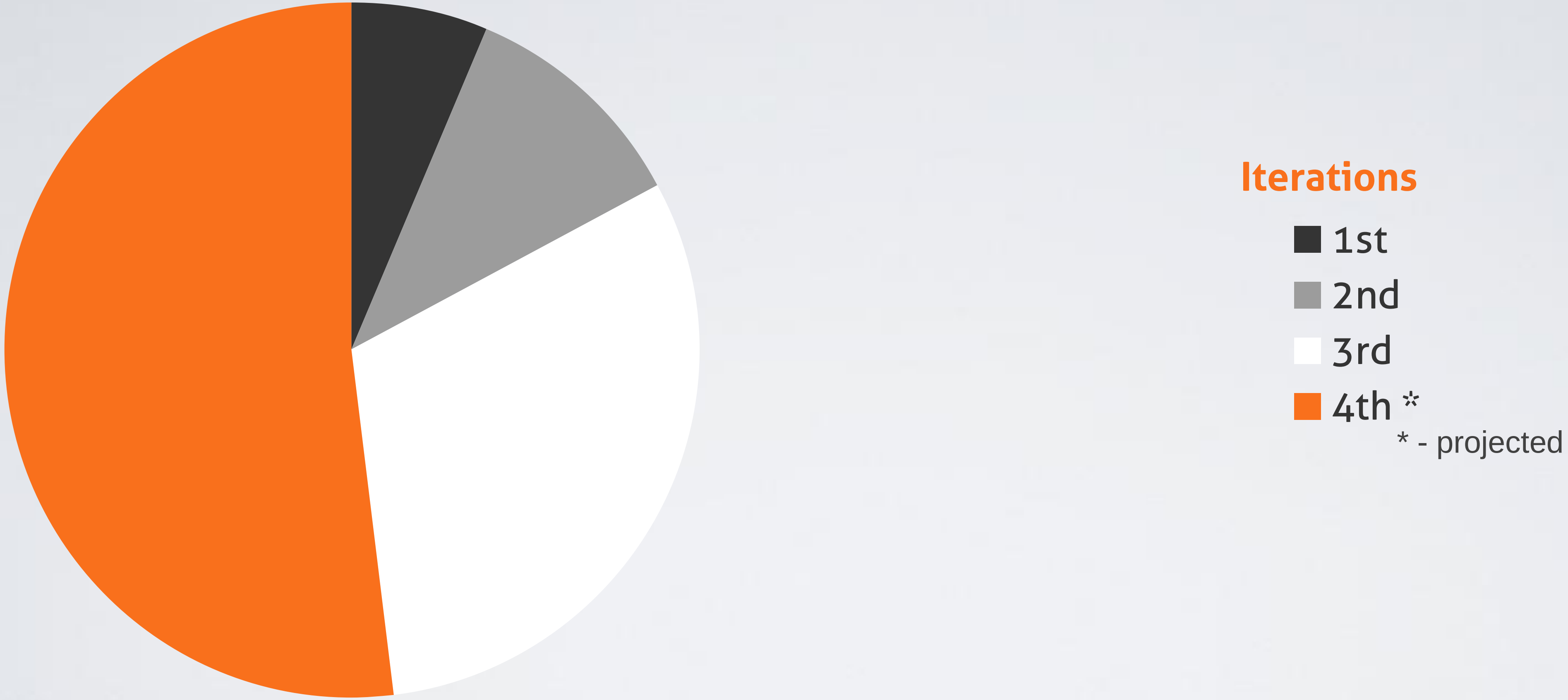
Development Time - Research

Increasingly productive



Development Time - Coding

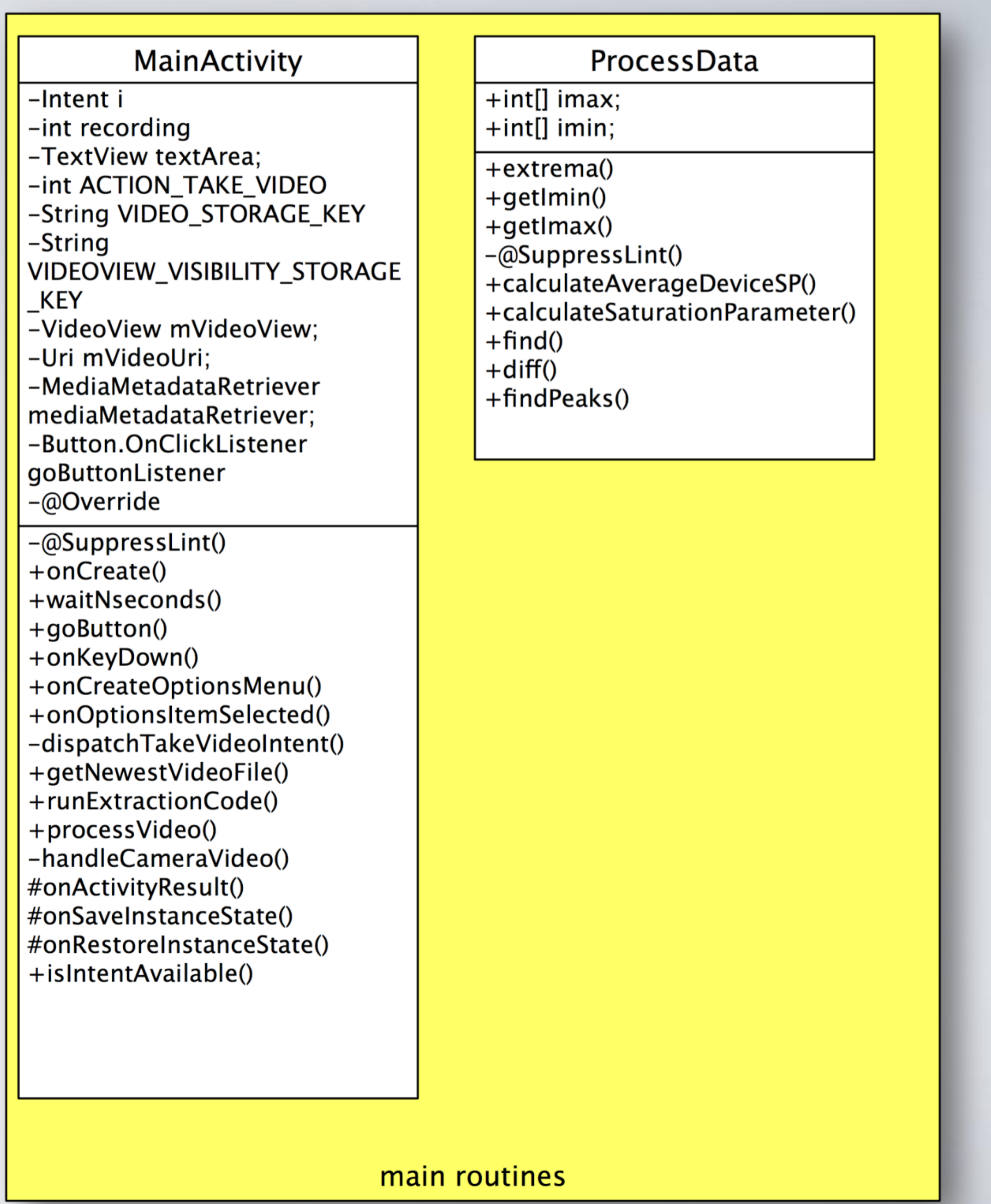
Increasingly productive



Code & Design Models

Main routine

Point of Entry for Functional Cases



MainActivity

-Intent i
-int recording
-TextView textArea;
-int ACTION_TAKE_VIDEO
-String VIDEO_STORAGE_KEY
-String VIDEOVIEW_VISIBILITY_STORAGE_KEY
-VideoView mVideoView;
-Uri mVideoUri;
-MediaMetadataRetriever mediaMetadataRetriever;
-Button.OnClickListener goButtonListener
-@Override

-@SuppressLint()
+onCreate()
+waitNseconds()
+goButton()
+onKeyDown()
+onCreateOptionsMenu()
+onOptionsItemSelected()
-dispatchTakeVideoIntent()
+getNewestVideoFile()
+runExtractionCode()
+processVideo()
-handleCameraVideo()
#onActivityResult()
#onSaveInstanceState()
#onRestoreInstanceState()
+isIntentAvailable()

ProcessData

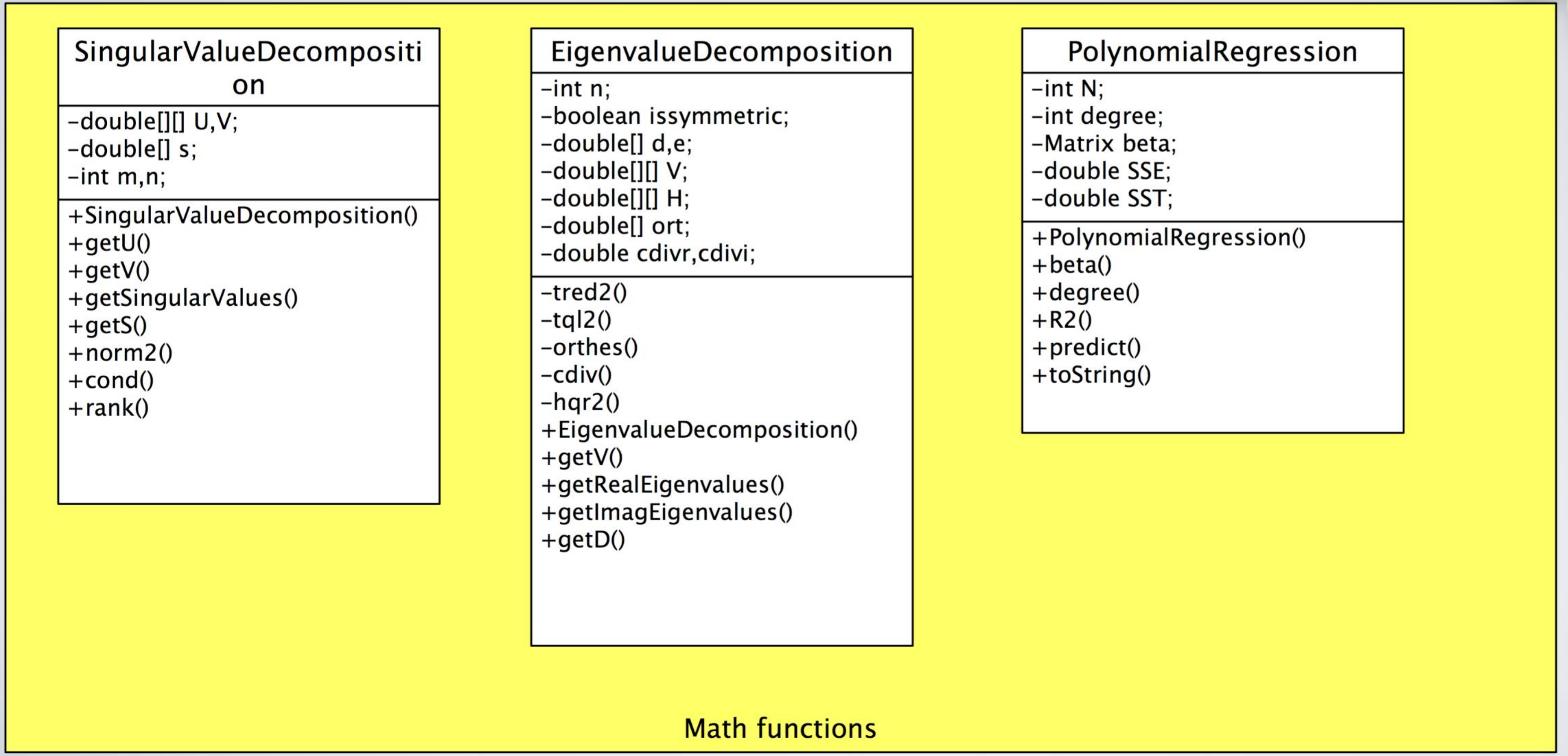
+int[] imax;
+int[] imin;

+extrema()
+getImin()
+getImax()
-@SuppressLint()
+calculateAverageDeviceSP()
+calculateSaturationParameter()
+find()
+diff()
+findPeaks()

main routines

Mathematical Functions

Processing Logic



SingularValueDecomposition

-double[][] U,V;
-double[] s;
-int m,n;

+SingularValueDecomposition()
+getU()
+getV()
+getSingularValues()
+getS()
+norm2()
+cond()
+rank()

EigenvalueDecomposition

-int n;
-boolean issymmetric;
-double[] d,e;
-double[][] V;
-double[][] H;
-double[] ort;
-double cdivr,cdivi;

-tred2()
-tql2()
-orthes()
-cdiv()
-hqr2()
+EigenvalueDecomposition()
+getV()
+getRealEigenvalues()
+getImagEigenvalues()
+getD()

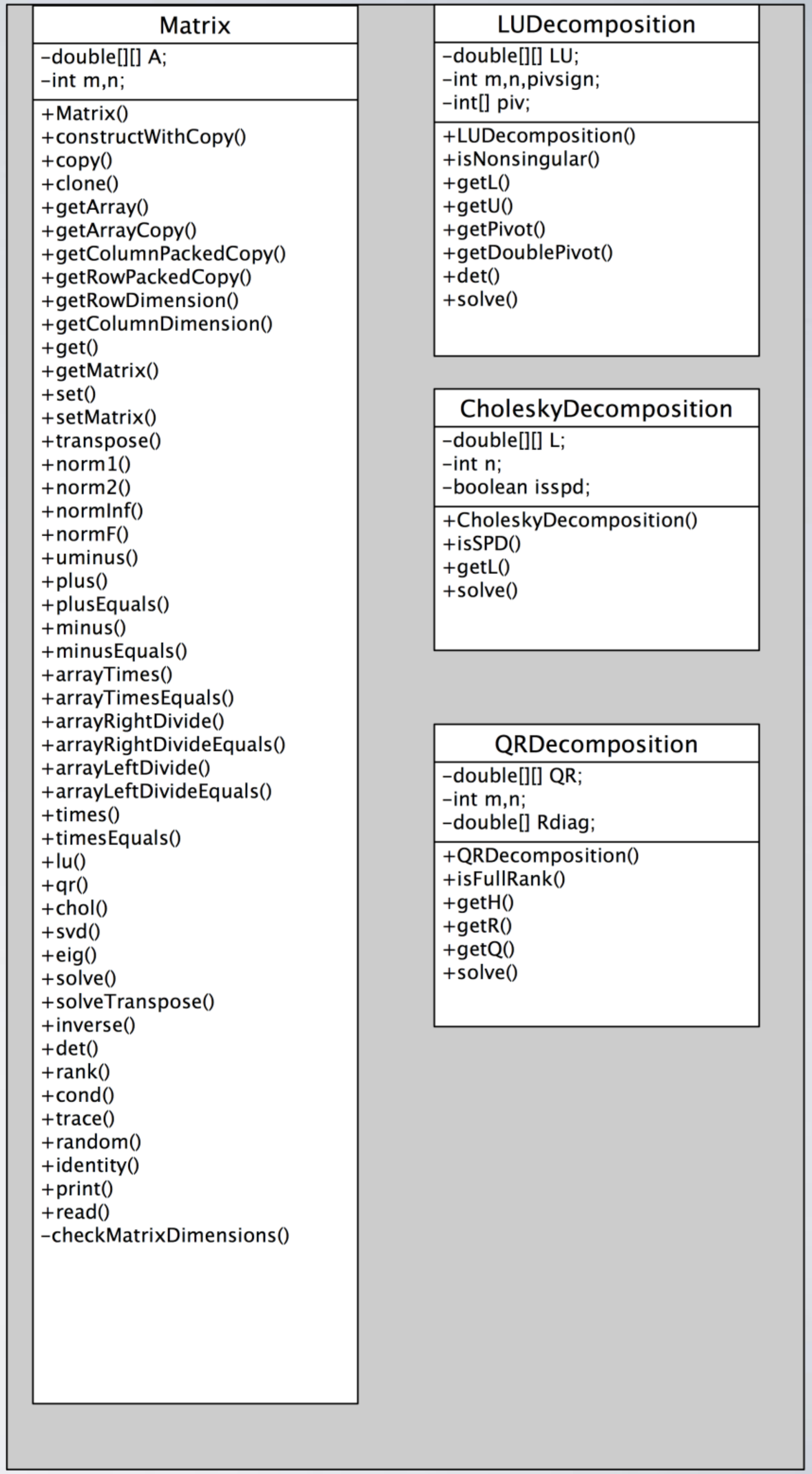
PolynomialRegression

-int N;
-int degree;
-Matrix beta;
-double SSE;
-double SST;

+PolynomialRegression()
+beta()
+degree()
+R2()
+predict()
+toString()

Math functions

Mathematical Functions Processing Logic



```
Matrix
-double[][] A;
-int m,n;
+Matrix()
+constructWithCopy()
+copy()
+clone()
+getArray()
+getArrayCopy()
+getColumnPackedCopy()
+getRowPackedCopy()
+getRowDimension()
+getColumnDimension()
+get()
+getMatrix()
+set()
+setMatrix()
+transpose()
+norm1()
+norm2()
+normInf()
+normF()
+uminus()
+plus()
+plusEquals()
+minus()
+minusEquals()
+arrayTimes()
+arrayTimesEquals()
+arrayRightDivide()
+arrayRightDivideEquals()
+arrayLeftDivide()
+arrayLeftDivideEquals()
+times()
+timesEquals()
+lu()
+qr()
+chol()
+svd()
+eig()
+solve()
+solveTranspose()
+inverse()
+det()
+rank()
+cond()
+trace()
+random()
+identity()
+print()
+read()
-checkMatrixDimensions()
```

```
LUDecomposition
-double[][] LU;
-int m,n,pivsign;
-int[] piv;
+LUDecomposition()
+isNonsingular()
+getL()
+getU()
+getPivot()
+getDoublePivot()
+det()
+solve()
```

```
CholeskyDecomposition
-double[][] L;
-int n;
-boolean isspd;
+CholeskyDecomposition()
+isSPD()
+getL()
+solve()
```

```
QRDecomposition
-double[][] QR;
-int m,n;
-double[] Rdiag;
+QRDecomposition()
+isFullRank()
+getH()
+getR()
+getQ()
+solve()
```

Contact me for the .apk

Thank You.