NEWS PHONE APPLICATION

Deliverables II

COP 4331 Section 1
Fall 2010

GROUP 8

Karl Banks

Aaron Birencwaig

Andrew Harmic

Jason Heintz

Stephen Rodriguez

Tyler Zaino

UNIVERSITY OF CENTRAL FLORIDA

TABLE OF CONTENTS

1 HIGH-LEVEL DESIGN

Modification history:

Version	Date	Who	Comment
v1.0	10/16/10	Andrew Harmic	Added High-Level Architecture
v2.0	10/21/10	Karl Banks	Added Design Issues

1.1 High-Level Architecture

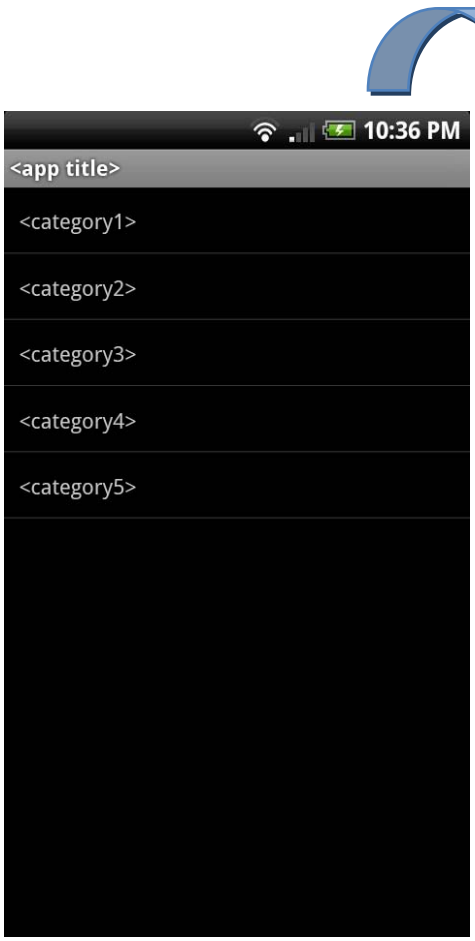


Figure 1 – Main Screen

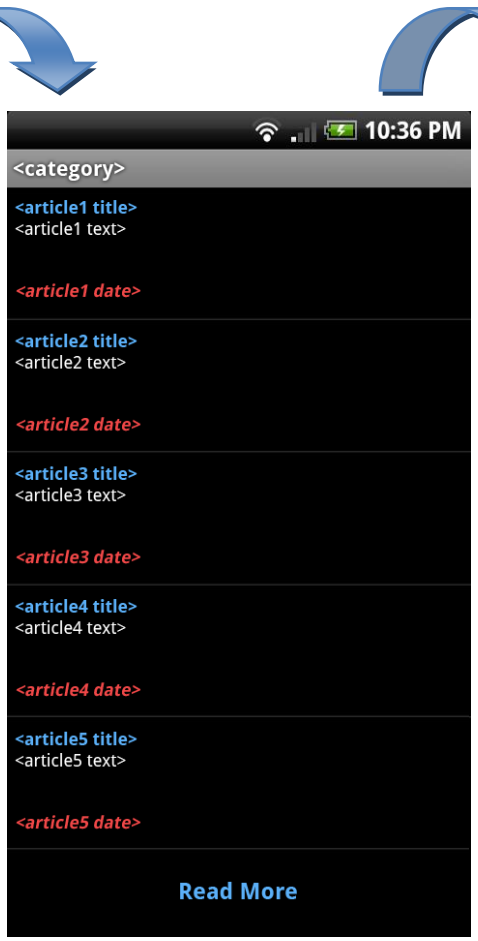


Figure 2 – Category Screen

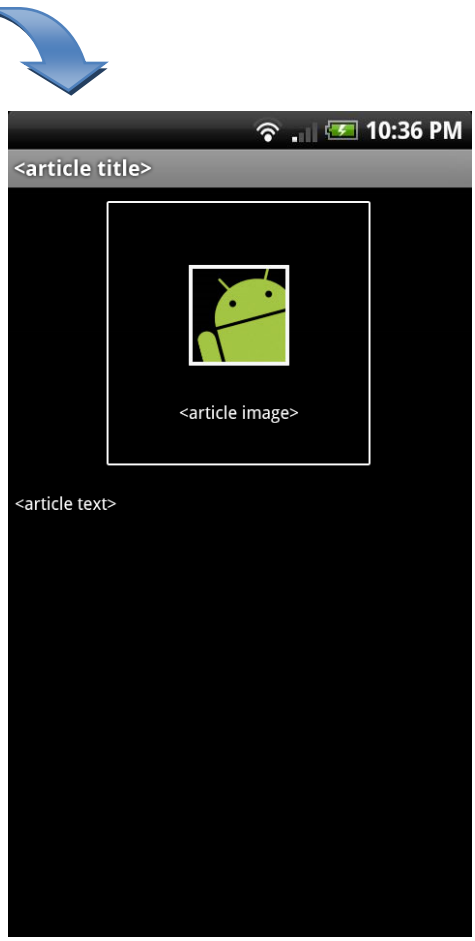


Figure 3 – Article Screen

The high-level architecture of the mobile news application can be broken into three main components, the "main screen", the "category screen", and the "article screen". The "main screen", shown in **Figure 1**, presents the user with the title of the mobile news application followed by a list of the five news categories. The "category screen", shown in **Figure 2**, presents the user with the five most recent articles posted in the selected category, as well as a "Read More" option which will load and display five additional articles upon selection. Each article will be formatted to display the article title (in blue), followed by a brief snippet of the article (in white), followed by the date and time the article was posted (in red). The "article screen", shown in **Figure 3**, presents the user with the selected article. This screen will display the article title followed by the article in its entirety, as well as an accompanying image in an arbitrary location. Navigation between these screens is denoted by the arrows between each figure shown above, allowing both forward (blue arrows) and backward (red arrows) movement. All data displayed on each screen will be provided live by the CNN news database, and will not be stored to the mobile device.

1.2 Design Issues

1.2.1 Reusability

The mobile news application will be scripted with reusability in mind. The key to making a program reusable is following good programming techniques, such as using constant variables (constants) which can quickly be modified, and building the project in small blocks such that the blocks can be rearranged or used individually in future projects. We will follow this "proper" practice when developing the application by taking full advantage of java's package/class hierarchy (the classes can be thought of as the "small blocks", because they can each be separately reused in future programs).

1.2.2 Maintainability

Due to the nature of this application, it will require little to no maintenance upon completion. Since the entire program is driven entirely by articles uploaded to the CNN news database, the core structure of the mobile application will not require updates on any sort of fixed-term basis. The only logical maintenance actions we can presently foresee would be adjusting the parsing algorithms to adapt to a new article format adopted by CNN. Should this occur, we will make the code very easy to modify by writing it cleanly (using appropriate tabs/whitespace) and frequently adding comments to it from the start.

1.2.3 Testability

The overall simplicity of the mobile news application shall allow for rigorous testing and fine-tuning. Since the system is comprised of only three basic interfaces, there is really only a small handful of scenarios that will affect the functionality of the program. Thus, the application is highly susceptible to testing, and will be subject to various test cases that will assess all of the aforementioned scenarios to ensure optimum and accurate performance. For more information on testing the mobile news application, see the "Test Plan" section (section 4) of the Deliverables I documentation.

1.2.4 Performance

To ensure maximum performance, the code will be written in the best style that supports the shortest execution time. Realistically, the only "executing" done by the mobile news application will be communicating with the CNN news database to retrieve article information and handle the data. With this, the performance of the program is really only limited by the performance of the mobile device and cellular connection. Also, writing the program to operate without storing any information will add to its high performance.

1.2.5 Portability

While it may sound redundant to mention that a *mobile* application will be portable, we must mention the fact that the application can be accessed in any given location. As long as the end-user is equipped with the required materials (mobile phone running Android operating system, internet access, etc.), they will have full access to the system. The application will be written in the Java programming language for Android platforms, meaning it will be universally accessible by any supporting Android cellular device. For more information on the required platform specifications, see the "Software Requirements Specification" section (section 3) of the Deliverables I documentation.

1.2.6 Safety/Security

The mobile application will not store any information inputted by the user, nor will it have access to any form of hazardous or sensitive material. Therefore, there are no major safety/security issues associated with the development of this project.

1.2.7 Prototyping

Software prototyping will be utilized throughout the development of the application by accessing the application on a real Android platform device. Each prototype generated will collectively demonstrate the various phases of development, providing additional functionality with each individual release. While these models may not actually be used to evaluate major alternative design strategies, they will be used to guide an appropriate formatting structure, indicating details such as font size, color, etc.

1.2.8 Technical Risks

The only major technical risk set forth by the mobile news application is its dependency on an outside source (the CNN news database). Should the news database go down or become unavailable, the program will be rendered useless as it will have no real data to populate with. Although this risk is high-impact and critical to the application, one may feel "reassurance" in knowing that the likeliness of such an event to occur is very minimal and to date has not yet occurred. Nevertheless, we will still treat this issue as a high-priority technical risk.

2 DETAILED DESIGN

Modification history:

Version	Date	Who	Comment
v1.0	10/17/10	Jason Heintz	Initial class diagram
v1.1	10/20/10	Aaron Birencwaig	Modified class diagram
v2.0	10/23/10	Tyler Zaino	Added detailed design issues
v3.0	10/24/10	Stephen Rodriguez	Added trace of requirements to design

2.1 Design Issues

2.1.1 Reusability

The program will be written with reusability in mind. This is a very beneficial practice for many reasons. The main reason is the availability to reuse the code for each section of news. This way we are able to maintain consistency in both the presentation and the content of each subsection of news. The practice of reusability also helps in the debugging phase. Since each subsection of news will be written with the same code, a bug found in one section will be easy to spot and fix in all of the other sections as well.

2.1.2 Maintainability

The way the application will be written makes maintainability a very easy task. Essentially, the program will scan through each section of the CNN website, and collect the news stories for the application. Once the new stories are found, the text is parsed and reprinted as an article on the application. The pictures associated with each article are also collected the same way. Unless CNN does a drastic overhaul of its website, the method that will be used by the application provides very little reason for the programmers to have to maintain it on a regular basis.

2.1.3 Testability

Due to the vast amount of reusability that will be built into the code, the testing of the application becomes much simpler. As stated previously, many of the bugs will likely be repeated throughout the application since the foundation of the code for each subsection will be the same. Also, the amount of different test cases will be at a minimum. This is a benefit because it allows the testing of each case to be very thorough and in depth to ensure each feature is working properly under all types of test conditions.

2.1.4 Performance

The code will be primarily written with performance in mind. For the application to be useful, it must run at peak performance for the majority of the time, because no one wants old news. This performance will be achieved by allowing the application to quickly retrieve the stories from the CNN website. This eliminates the need to store the articles in a database. This method cuts down the amount of overhead it would cost to consistently access the database, which at peak times could be flooded with users, slowing down the retrieval of the news articles.

2.1.5 Portability

The portability of the application is also a main concern during the development of the design. The essential purpose of a mobile application is for it to be a scaled-down version of the actual website, providing only the essentials to the user's phone. By design, the application will be written with well organized subsections that allow the user to quickly access the top news stories they desire. If the user wants to continue to look for new articles, an option is available to view more. This allows for the application to be compact and efficient. Also, the code will be written to allow any Android device to use the application. This will be done to ensure the majority of the Android users have the availability to use our application.

2.1.6 Safety/Security

To eliminate much of the need to add security features, the mobile application won't be programmed to hold any of the user's personal or confidential information. Also, since the news stories will be taken directly from the CNN website in a text format, there is very little risk of infection by virus. The only safety measure that will be built into the code is the way the program parses the stories from the website. The text parser must be very dynamic to ensure no unwanted text will able to pass through to the application.

2.1.7 Prototyping

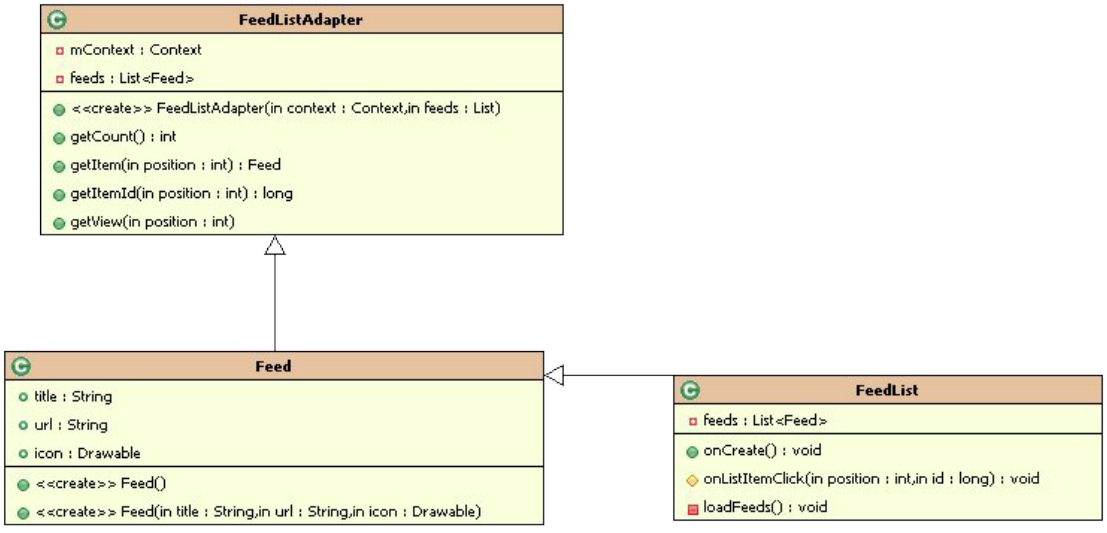
The program will be written with many subsections, allowing for the release of many prototypes. The initial prototype will be the basic interface the user will encounter each time the device is loaded up. Then after each section is individually created, a new prototype will be released with that function now enabled. This will allow for testing to begin quickly. Also, releasing the application in pieces will allow for regression testing, this is used to ensure that each new release does not affect the functionality of the previous releases.

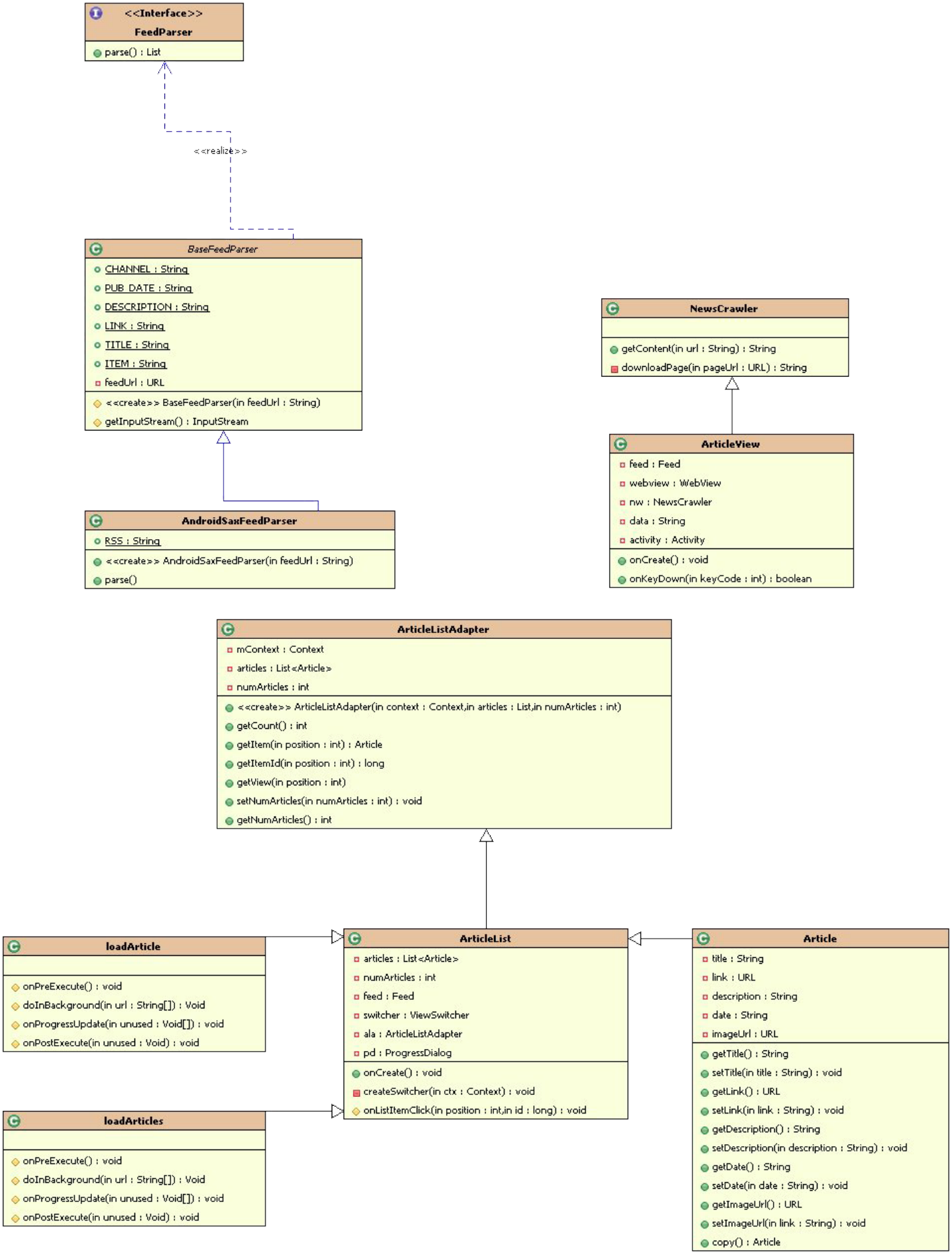
2.1.8 Technical Risks

To yield better performance, there is always a sacrifice. In this application, the benefit outweighs the single risk. Due to the nature of the program, the application will rely heavily on the CNN website to gather its stories. Since a database is going to be eliminated from the design to boost performance, if the CNN website goes down, the news stories will be unavailable until their site is back up and running. Though this may be seen as a critical risk, it will also ensure that all of the news stories being provided by the application are up-to-date.

2.2 Detailed Design Information

2.2.1 Class Diagram




<<Interface>>
FeedParser
parse() : List
<<realize>>
BaseFeedParser
CHANNEL : String
PUB_DATE : String
DESCRIPTION : String
LINK : String
TITLE : String
ITEM : String
feedUrl : URL
<<create>> BaseFeedParser(in feedUrl : String)
getInputStream() : InputStream
AndroidSaxFeedParser
RSS : String
<<create>> AndroidSaxFeedParser(in feedUrl : String)
parse()
NewsCrawler
getContent(in url : String) : String
downloadPage(in pageUrl : URL) : String
ArticleView
feed : Feed
webview : WebView
nw : NewsCrawler
data : String
activity : Activity
onCreate() : void
onKeyDown(in keyCode : int) : boolean
ArticleListAdapter
mContext : Context
articles : List<Article>
numArticles : int
<<create>> ArticleListAdapter(in context : Context,in articles : List,in numArticles : int)
getCount() : int
getItem(in position : int) : Article
getItemId(in position : int) : long
getView(in position : int)
setNumArticles(in numArticles : int) : void
getNumArticles() : int
loadArticle
onPreExecute() : void
doInBackground(in url : String[]) : Void
onProgressUpdate(in unused : Void[]) : void
onPostExecute(in unused : Void) : void
loadArticles
onPreExecute() : void
doInBackground(in url : String[]) : Void
onProgressUpdate(in unused : Void[]) : void
onPostExecute(in unused : Void) : void
ArticleList
articles : List<Article>
numArticles : int
feed : Feed
switcher : ViewSwitcher
ala : ArticleListAdapter
pd : ProgressDialog
onCreate() : void
createSwitcher(in ctx : Context) : void
onListItemClick(in position : int,in id : long) : void
Article
title : String
link : URL
description : String
date : String
imageUrl : URL
getTitle() : String
setTitle(in title : String) : void
getLink() : URL
setLink(in link : String) : void
getDescription() : String
setDescription(in description : String) : void
getDate() : String
setDate(in date : String) : void
getImageUrl() : URL
setImageUrl(in link : String) : void
copy() : Article

2.3 Trace of Requirements to Design

No: 001
Statement: The News Phone Application shall display content on the following five areas: US, World, Money, Entertainment, and Sports
This is handled in multiple areas. First these sections are displayed by the drawable class for the user interface. Then the feed class gets the articles based on the type of the content.
No: 002
Statement: The News Phone Application shall depict a single important headline news item at its home screen.
This will be handled in the Article class by the getImage function.
No: 003
Statement: The News Phone Application shall have a drop down available that lists all news sections covered.
This will be handled by the drawable class that will provide the interface with all the subsections listed.
No: 004
Statement: The News Phone Application shall display, at the minimum, five news items per news section covered.
This will be handled in the ArticleListAdapter class by the setNumArticles and getNumArticles functions.
No: 005
Statement: The News Phone Application shall give the user the ability to show more news articles for each news section covered.
This will be handled in the ArticleListAdapter class by the setNumArticles and getNumArticles functions by changing the increasing the number of articles.
No: 006
Statement: The News Phone Application shall display an error message whenever the application fails to connect to the internet.
This will be handled in the ConnectivityManager class by the extra_no_connectivity function.
No: 007
Statement: The News Phone Application shall display content for every link within five seconds upon execution of that link.
This will be handled in the articleList class and the performance will stem from the fact the articles are coming directly from the CNN feed.