NEWS PHONE APPLICATION

High-Level Design

COP 4331 Section 1
Fall 2010

GROUP 8

Karl Banks

Aaron Birencwaig

Andrew Harmic

Jason Heintz

Stephen Rodriguez

Tyler Zaino

UNIVERSITY OF CENTRAL FLORIDA

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 High-Level Architecture

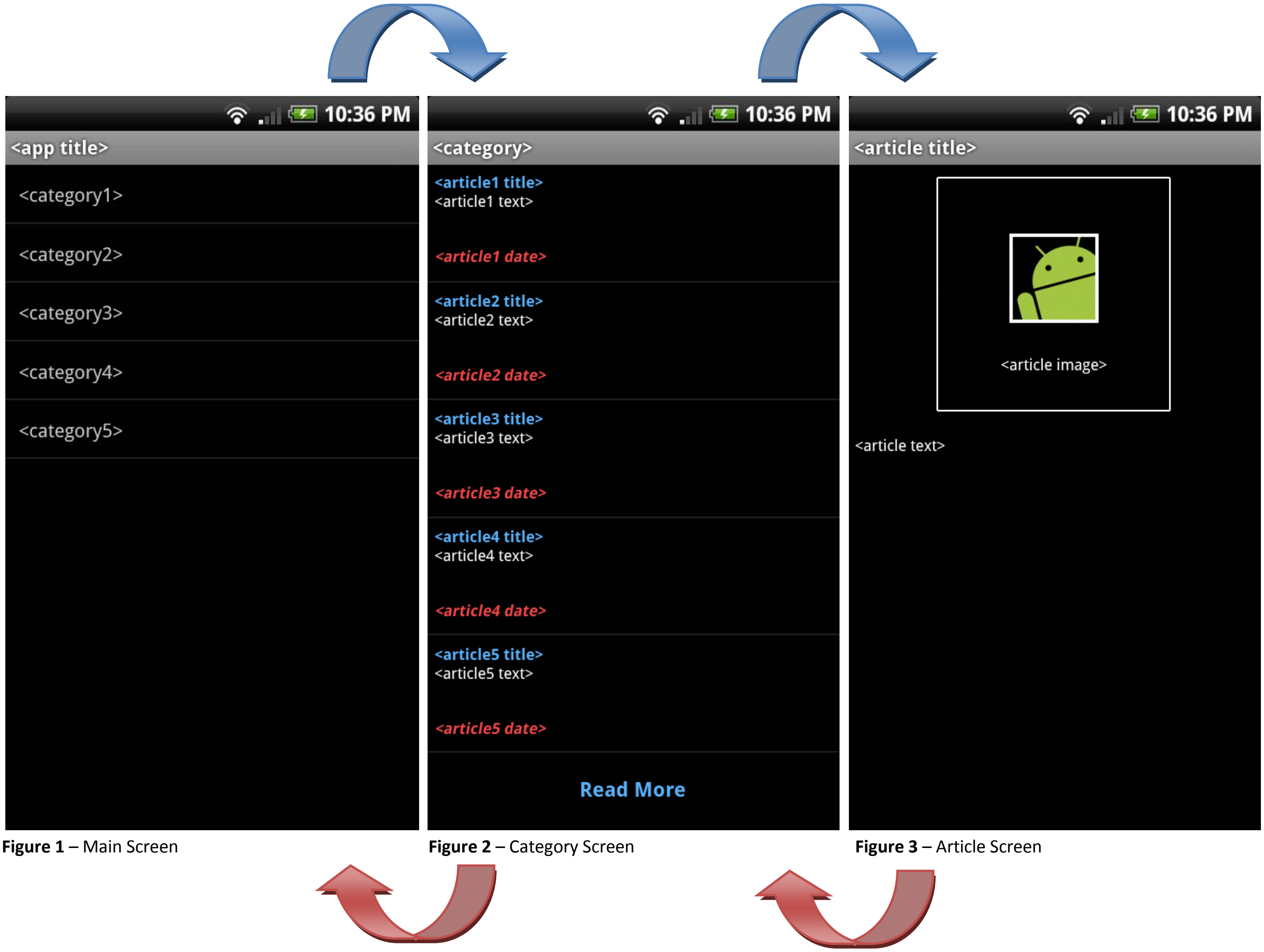

Figure 1 – Main Screen

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.

2 Design Issues

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).

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.

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.

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.

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.

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.

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.

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.