Daniel Alabi, Amy Becker, and Michael Domingues
Final Project Proposal
May 26, 2011
Knight Reader

OVERVIEW
(see attached mockups)

Knight Reader is a feed reader that allows users to organize and manage their RSS feeds. The program displays a user's subscriptions (feeds) as well as the stories within these feeds. Furthermore, the program allows for searching within the stories and sorting by various parameters. Consider the following scenarios, which demonstrate features of the program:

- Ellie enjoys reading comics. Every morning she uses the feed reader to view the new daily strip of her favorites. Recently, her friend recommended that she start reading the xkcd comic. She simply clicks "Subscribe" and enters the url into the text field on the window to subscribe to the new comic, knowing that she can delete it later if she wants. She can choose to add xkcd to her 'Comics' folder, so that it will be listed among her other comics subscriptions, or she can add it to a different folder (or even to a new folder). She can click on the feed title to see all of the recent 'stories' (comics) and revisit previous entries as she likes.

- James is already closely following the 2012 presidential election. He is subscribed to twelve different RSS news feeds offering a full range of political slants and perspectives. He frequently makes use of the reader's search bar, checking keywords like "presidential candidate" or "2012 election" to see all of the recent stories of interest, from all of his feeds. He can quickly see which stories are unread because they are listed in bold on the story list. Also if his feeds are being frequently updated, James can manually "Fetch" the feeds to grab new stories if the automatic updater is not frequent enough. Furthermore, he can switch to the 'Browser view' to peruse the actual website corresponding to the feed or referenced within a given feed story.

DESIGN PATTERNS

Knight Reader will make use of several design patterns. Our main pattern is the Model-View-Controller pattern. The view component is our user interface, which displays the feeds and stories to the user in a clear way. Specifically, the view maintains two different display types, the 'story-list' view and the 'browser' view and uses information from the model to fill these displays. The model manages the data about the users subscriptions and the stories for each feed. The view can access the model's methods in order to display specific information to the user. The controller also communicates with both the view and the model, based on user input. For example, the controller responds to the user commands to sort and search, and these options are reflected in the display (view). Also, the controller manages user input to subscribe to new feeds and to fetch feeds, resulting in changes to the model itself.

Secondly, we will use the Singleton pattern for the KnightReader class itself. We do this because we want to limit the program to one instantiation. We don't want multiple readers open at the same time, as they may contradict each other.

Thirdly, we make use of the Observer pattern. Since the content displayed in the reader is constantly being updated, we need these changes to be reflected to the user. Therefore, our model acts as a subject and the view is an observer that responds to changes in the model.

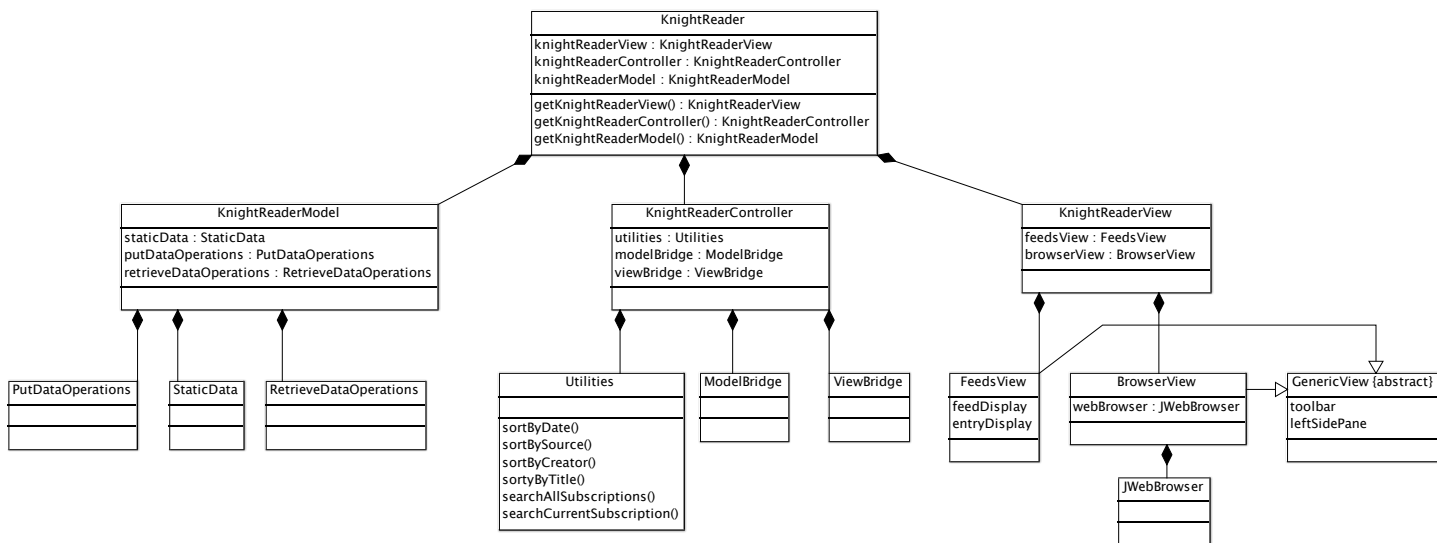UML/CLASSES
      (see attached UML diagram)

      The architecture of our class structure reflects the MVC pattern.   The main class, KnightReader, has an attribute and a corresponding method for each of the model, view and controller. The Model contains the static data, as well as operations for changing and retrieving this data. The View maintains two subclasses of the abstract class GenericView, namely the BrowserView and the FeedView. Finally, the Controller contains utilities that control operations like sorting and searching, as well as bridges to the other two branches (view and model).

DATA FILES AND DATA SOURCES
- Program data will be stored in a data directory, in which each feed subscription has a corresponding file which contains information about the feed's stories (read vs unread) and about the feed itself.
- The sources of our data are actual RSS feeds, which the user specifies and subscribes to. The specific sources are therefore dependent on user preferences.

OTHER RESOURCES

      We will be making use of feed4j, which is a feed parser library for Java, as well as JWebBrowser from DJ Native Swing, which will allow us to create our browser display.

KnightReader

knightReaderView : KnightReaderView
knightReaderController : KnightReaderController
knightReaderModel : KnightReaderModel

getKnightReaderView() : KnightReaderView
getKnightReaderController() : KnightReaderController
getKnightReaderModel() : KnightReaderModel

KnightReaderModel

staticData : StaticData
putDataOperations : PutDataOperations
retrieveDataOperations : RetrieveDataOperations

KnightReaderController

utilities : Utilities
modelBridge : ModelBridge
viewBridge : ViewBridge

KnightReaderView

feedsView : FeedsView
browserView : BrowserView

PutDataOperations

StaticData

RetrieveDataOperations

Utilities

sortByDate()
sortBySource()
sortByCreator()
sortyByTitle()
searchAllSubscriptions()
searchCurrentSubscription()

ModelBridge

ViewBridge

FeedsView

feedDisplay
entryDisplay

BrowserView

webBrowser : JWebBrowser

GenericView {abstract}

toolbar
leftSidePane

JWebBrowser

FEED VIEW:

| File | Edit | View | About | ✕ |

Fetch Feed **Fetch All Feeds** **Subscribe**

keyword **Search**

📁 **News**
- ◆ CNN — (1)
- ◆ NPR News — (7)
- ◆ NYT> Science — (4)
- ◆ MacWorld — (2)

📁 **For Fun**
- ◆ Yahoo! Sports—MLB — (12)
- ◆ Pearls Before Swine — (2)

🗑

Pearls Before Swine	**Date**
Comic for May 26, 2011	5/26/2011
Comic for May 25, 2011	5/25/2011
Comic for May 24, 2011	5/25/2011
Comic for May 23, 2011	5/23/2011
Comic for May 22, 2011	5/22/2011
Comic for May 21, 2011	5/21/2011
Comic for May 20, 2011	5/20/2011

Comic for May 24, 2011

BROWSER VIEW:

| File | Edit | View | About | ☒ |

Fetch Feed **Fetch All Feeds** **Subscribe**

keyword   **Search**

📁 **News**

- CNN   (1)
- NPR News   (7)
- NYT> Science   (4)
- MacWorld   (2)

📁 **For Fun**

- Yahoo! Sports—MLB   (12)
- Pearls Before Swine   (2)

🗑

NYT> Science "New Mileage Stickers..."   **Back to Main**