Comet Reader

Project Plan

Contracted By Cognitive Thought Media

Table of Contents:

Description	**Page Number**
1.0 Project Overview	3
2.0 Project Life Cycle	4
3.0 Detailed Design	5
4.0 Quality Management	6
5.0 Risk Register	7
6.0 Schedule Management	8

1.0 Project Overview

1.1 Project Overview - Description

Comet Reader is an RSS reader for Android that provides the user with the ability to view RSS and ATOM feeds from the internet. Individual feeds are displayed in chronological order and the user has the option to view all of their feeds at once or individual feeds. The user can also select to view all posts, unread posts, read posts, or starred posts. When a user selects a post it is marked as read and an extended preview of the post is shown. The user then has the option to star the post, share it on social networks or by email, or to open it in a web browser.

1.2 Detailed Project Description

Comet Reader is designed to provide both an intuitive and fully functional interface to view content from RSS and ATOM feeds. It was designed from the ground up for Android and takes advantage of the latest design standards including Google's new material designed, the Action bar, and sharing and web intents. Comet reader has been design to adapt to any screen size in both landscape and portrait mode and works on almost any Android device running at least Android version 4.0 (ice cream sandwich). Users are given the option to input their own feeds by supplying the URL for that feed, the URL of any Feedburner page, and in some cases the URL of a website if it contains a link to an RSS feed. Comet Reader will parse the feed into individual posts which will be displayed in chronological order in the user's main feed.

Each post will be represented using an image pulled from the RSS feed if available, the title of the post, a preview of the description or content of the post, and the name of the post. These posts will show up in a user's content feed and can be scrolled through. Posts from all of the user's feeds can be viewed together or the user can select to only view posts from specific feeds. The user can also select to only view all posts, read posts, unread posts, or starred posts. By selecting a post a user is taken to the post view.

In the post view the user sees a larger image taken from the post as well as the title of the post and the full description or content of the post. Any HTML formatting or images embedded in the post will also show up on this page. The user can scroll through the post's content or description and at the bottom of the post is given the option to open the post in a web browser. In the action bar the user has the option to star the post for easy access later and to share the post to almost any social network or email client that the user has installed on their device using share intents. When the user first selects the post it is marked as read and consequently the title will not appear bold in the user's main feed.

2.0 Project Life Cycle - Evolutionary Prototyping Methodology

The project will be developed in accordance with the evolutionary prototyping model. In this model a prototype will be developed and tested and then revised into another prototype until a finished app is produced. The stages in the model are:

● Analyze Requirements- The requirements of the project will be analyzed to ensure that development will be done in accordance with them .
● Build Prototype - A working prototype will be produced.
● Test Prototype - The prototype will be tested for both design and bugs both by myself and by others.
● Get Feedback - Any bugs or suggestions will be recorded to be addressed in the next prototype.
● Evaluate Prototype - The prototype will be evaluated to determine what issues still need to be addressed and what is ready to be finalized.
● Finalize App - All features in the app will be complete and finalized. Development will be finished.

Lifecycle Diagram - Evolutionary Prototyping Methodology

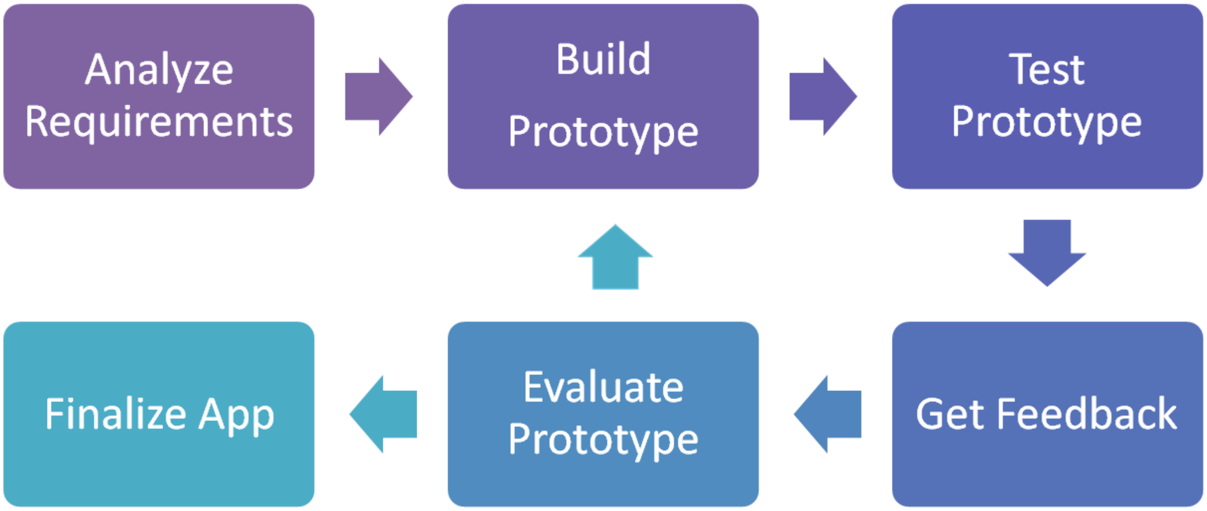

Fig. 2.1

3.0 Detailed Design

Comet Reader will be designed to provide users with a simple and intuitive experience while also giving them access to all the functionality that they would expect in an RSS reader. I created this data flow diagram to map out the movement of a user through the app. The design centers around two screens. The first is the Main Feed Page which shows a chronological list of feeds from a user's RSS subscriptions. The user can use two spinners to select to view all feeds or only a specific feed, to add new feeds, and to view only read, unread, or starred posts. On this page each post will be represented with a title, text preview, image if available, and the source of the poset. The second page is the post page which shows a larger image pulled from a post, the title, a longer preview of the content, more images, if available, and buttons to open the post in a web browser, star the post, and share the post.

Data Flow Diagram

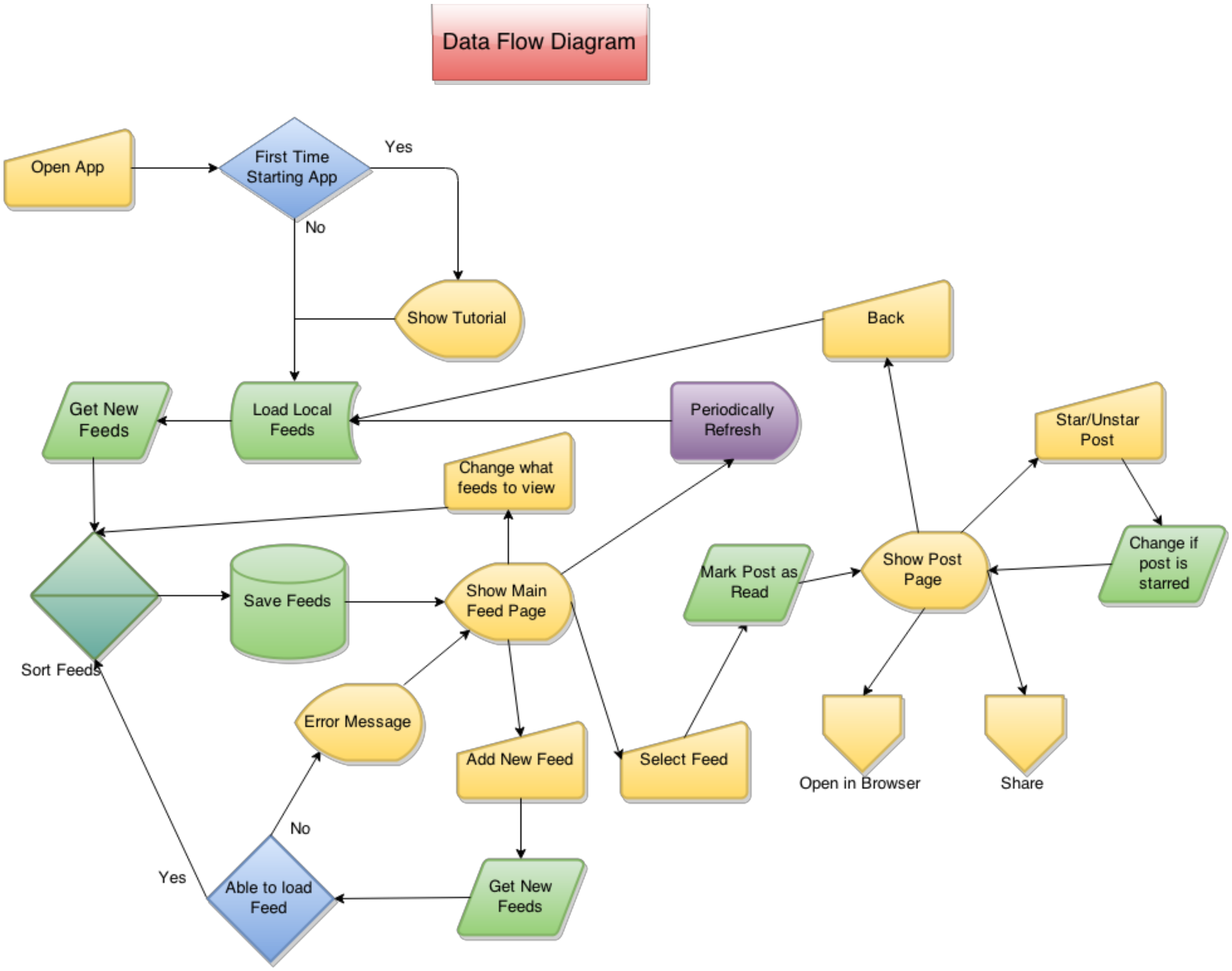

Fig 3.1

4.0 Quality Management:

One advantage to the evolutionary prototyping methodology for software development is that it allows for frequent quality checks through its cycle of continual prototyping and testing. To ensure that the app is both functional and intuitive I will frequently demo new prototypes to friends, family members, and fellow students and have have both others and myself test them. As Android devices come in many different screen sizes and aspect ratios, I will test the app on a variety of devices in both landscape and portrait mode to ensure that it is both functional and intuitive on all screen sizes. I will keep a log of all bugs found and suggestions given when testing the app and will attempt to implement and fix as many as possible in each new prototype.

5.0 Risk Register:

Risks are inevitable in any large project. Consequently, this register documents and discusses the mitigation of several possible risks and concerns.

Risk Description	Consequence	Mitigation and Recovery
Lack of familiarity with the Android Studio development environment	Development may proceed slower than expected as I have to learn to use the development tool at the same time I develop the app	Extra time has been allotted at the beginning of the timeline to familiarize myself with the development environment as well as additional time near the end of the timeline in case development falls behind schedule.
Inability to parse RSS feeds or implement other required features	This would result in the project not meeting the requirements or function	Extra time has been allotted at the beginning of the schedule to account for learning to implement features that I lack experience implementing. Additionally the first milestones are scheduled far enough from the final submission date that I should have time to learn how to implement all the required features.
The project falls behind schedule due to unforeseen events either in development or in my life	The app might not be finished in time	Extra time has been allotted near the end of the schedule to account for any delays and I have set milestones to help me keep development on track.

Fig. 5.1

6.0 Schedule Management

Development will adhere to these milestones to ensure that development will remain on track:

Date	Description
11/1/14	Research into design and development will have begun. I will begin determining the specifics for what tools will be used and how RSS and ATOM feeds are parsed.
12/1/14	Research into the tools needed and design specifics will be finalized. I will be in the process of developing the first working prototype of the RSS reader.
12/19/14	The first prototype will be complete and will have been tested and shown to others to gain feedback. Work on revising the prototype will begin.
21/26/14	Additional prototypes will be produced and the basic user interface will be complete. The user should now be able to add their own RSS feeds and view the content both in a preview and in a web browser.
1/2/15	More prototypes will have been produced and revised. The basic user interface and functionality will be complete. Work will have begun on adding additional features and documentation.
2/1/15	All features will be implemented and finalized. I will begin making the final revisions to the code and documentation.
2/22/15	All materials will be finalized and ready for submission.
2/23/15	Final deadline for submitting all materials.

Fig. 6.1