

Initial Plan – News Reader with Kill File

Author: Hani Chaaban

Supervisor: Professor Ralph Martin

Module: CM3203 One Semester Individual Project – 40 Credits

Project Description

This project centers on the development of a News Reader application with a Kill File, which the user can add keywords to. These keywords will be treated as topics to be ignored, allowing them to cultivate a news feed which is tailored to their interests as well as ignoring news which they deem irrelevant.

The application will be primarily targeting the Android platform, with extended scope allowing for additional challenge through enriched features and code reuse to target other platforms, devices form factors and operating systems.

The application will leverage RSS/Atom feeds for content on a basic level – short articles with minimal rich content. The potential for scraping longer length, richer content from the source (rather than the RSS feed) on the fly (with images/videos and more extensive content) will be investigated, and if possible within the timeframe of the project, implemented.

Project challenges will arise based on the rigor and professionalism of my development approach, the application's aesthetic and usability, as well as from additional feature development and personal learning aims, which I've outlined below and in the following sections.

Project Aims

Based on some preliminary research around the project, I have come up with a set of personal aims. Primarily, these will aid me in designing and developing professional applications for a modern OS in the future, by furthering my knowledge in the following areas through research and practical use:

- Creating, understanding and applying a modern development cycle methodology to meet time constraints and feature requirements
 - o My chosen method for the project's development cycle is one which I learned during my placement year at Microsoft. I will refer to it in this and future reports as the Iterative Development Cycle.
 - o This method can be summarized as a combination of Kanban, Agile and iterative development concepts. Functionally, it prioritizes meeting a "Minimum Viable Spec" – the bare minimum requirements of the project, and then augmenting this in an iterative fashion by "stack ranking" the importance/difficulty ratio of any additional feature ideas, and developing them in a "Timeboxed" fashion.
- Modern UX design and the application of Human Computer Interaction principles
- User state synchronization and management via Cloud Computing tools and APIs
- App distribution and meeting publication requirements

These aims, and how I plan to achieve/leverage them, are outlined in further detail in the work plan below.

Minimum Viable Spec. Objectives

These objectives outline the basic requirement list that must be met to satisfy the application's feature requirements as described in the project description. I refer to this as the Minimum Viable Spec. These features and tasks must be complete for the application to be deemed functional, and once they are complete they must be re-tested to ensure additional "nice to have" features do not compromise the minimum viable spec.

I plan to automate this testing if possible using Unit tests/Test driven development, but some preliminary research is needed before this can be guaranteed, as it is contingent on which language, framework and APIs/Tools I select in my ongoing pre-development research analysis.

- Pre development research
 - o Further research, analyze and document relevant similar apps and platform-typical design conventions to enable wireframe planning & use case / user profile development
 - o Analyze benefits/downsides of different design approaches by applying my Human Computer Interaction knowledge & research / assess how they impact an application's performance and usability.
 - o Wireframe basic UI ideas & Use cases / User profiles.
 - o Evaluate potential development tools/frameworks that I can apply in development, and justify choice of language, API's and libraries to use for implementation. (e.g. cloud APIs for account synchronization)
- Develop Basic App Functionality incrementally
 - o Basic (functional only) UI
 - o Allow user to "subscribe" to RSS feeds
 - o Gather RSS Feed data and display it to user
 - o Allow user to filter feed data using a "kill file" to exclude undesired topics
 - o Allow user's state (articles, subscriptions etc.) to Synchronize to Google Account
 - o Allow article to be opened in the user's default browser
- Research and design Modern UI
 - o Should be aesthetically appealing, and apply Material Design principles on Android
 - o Develop multiple prototypes (mocks/wireframes) and collect user feedback to reinforce and iteratively improve design choices during development
 - o Develop branding to accompany app, potentially field test various logos, names etc. with a focus group if time permits

Bonus Objectives/Features – “Nice to haves”

This section documents ideas I have for feature improvements to the app or for further development and research, but which fall outside of the basic app requirements.

Under the Iterative Development Cycle methodology I have chosen, features which do not compromise the minimum viable specification when absent are part of the evolutionary phase of the development cycle, and prioritization/planning of development time for these features occurs after the previous time box is complete, as outlined in my Work Plan. This, along with other aspects of the Iterative Development Cycle methodology I am following, will be described and analyzed in further detail for its successes and shortcomings in my final report once the project is nearing completion, and I will compare it to other popular development cycles such as the Waterfall method.

As such, once the minimum spec and Modern UI have been developed, the following items will be stack ranked and prioritized based on development effort, feature value and development time remaining, with the aim of maximizing the app's feature list and user appeal and stability. This will occur at the end of each Timebox, as shown in my work plan.

- **Cross Platform App: Code reuse/Shared codebase**
 - o I would like to explore the potential for code reuse between multiple OS's and device types, potentially leveraging new and interesting frameworks like Apache Cordova^[1] and the Xamarin platform^[2]. I will analyze the cost/benefit of a universal cross platform development approach, including initial development effort, and ongoing maintenance cost delta, and multiple-platform release benefits.
- **App Feature improvement / extension**
 - o **Kill File Semantics:** The topics in the kill file could be generalized to provide a better semantic match- inferring the user's intended topic more accurately. For example, a basic kill file might perform a word search for the topic to ignore (i.e. if “iPhone” is in kill file, ignore articles containing “iPhone”) but a more advanced one might employ Regex or another generalization tool to also ignore “iPhones”, “iphone”, and other such variations of the topic. I will research and outline potentially improved solutions and potentially select a reasonable implementation to develop, if it fits my development timeline.
 - o **Account synchronization Convenience:** Account synchronization could be made more convenient to users by allowing synchronization to other account types (Twitter, Facebook, Microsoft account...).
 - o **Content Accessibility:** Content might be made more readily accessible by fetching and caching content for offline use on a user determined interval, allowing for reading in situations without an internet connection (for example, on the tube).
 - o **Adaptive UX :** While pursuing cross platform development, each platform should have a Modern and similarly featured UX, but also be differentiated in style. Researching and implementing custom UI designs based on screen size/OS/Device type would maximize usability for a wider range of users.
 - o **Subscription Grouping & Management:** Categorization and grouping of the user's subscriptions might allow for easier feed management, but requires additional thought & design for a management UI.
 - o **Reading Queue:** In my preliminary research, I have found that having the ability to queue items for later reading or “favourite” articles and share them are staple features in many popular readers and even upcoming web browsers^[3], and as such it would be nice to include them for feature parity and futureproofing.
 - o **App Store Release:** Through research of app store guidelines and (time permitting) performing the modifications needed to allow actual submission.

Work plan

This section outlines my estimated timelines for each goal/task to complete, as well as a basic overview of how I intend to achieve the task.

Due to the stack ranking method for the bonus objectives dictating when and how quickly they will be completed, and the order being undetermined until development starts, I have listed a generic "Bonus Objectives: (Number)" in places where the development on these tasks will be occurring. My final report will contain a revised work plan showing which specific features evolved at each point and the justifications behind that choice.

Weeks 1-2: Pre Development Research & Wireframes

- Download other readers and investigate their benefits / shortcomings to avoid during my development
- Find and investigate usability of various frameworks and toolkits that may aid my app, for example an image album viewer to display large collections of images that may appear in articles
- Choose a programming language (Likely Java, C# or HTML/CSS) based on useful frameworks/toolkits available for parsing RSS, scraping rich content (images/videos) etc.
- **Deliverable:** A non-functional, basic UX wireframe app should be ready by the end of this period to show the user flow through the app and the locations of various features in the UI
- **Deliverable:** Notes on the above research and decision justifications thus far.
- **Deliverable:** Use cases / User profiles & notes regarding their impact on the Wireframe's navigation/layout flow.
- **Supervisor review meeting**

Weeks 3-4: Basic app functionality & Modern UI design

- The basic UX wireframe app will be augmented iteratively with features that make up the minimum viable spec:
 - o Subscribe to feeds
 - o Get & Show RSS Data from subscribed feeds
 - o Add and filter topics using Kill File
 - o Synchronize user data to Google account
 - o Allow article to be opened in browser
- Outline and document a set of designs for advanced branding/UX themes (app name, logo, colour themes etc.)
- **Deliverable: Basic Minimum Viable Spec App with the above features.**
- **Deliverable: Selected brand and justifications/focus group data for each design.**
- **Supervisor review meeting**

Work plan (continued)

Weeks 5-6: Modern UI implementation

- Basic UX wireframe app will have assets replaced and tweaked based on selected brand and design.
- Research and document Modern UX considerations for OS (Material Design on Android^[4]) & create brief notes on how to apply to my app
- Basic UX wireframe app will be altered to fit OS guidelines for Modern UX based on research notes.
- **Deliverable: New version of app with Material Design/Modern OS design notes implemented**
- **Deliverable: notes on documentation of modern UI and decisions I made on how to apply them to my app**
- **Supervisor review meeting & Next Timebox planning for bonus objectives/features**

Weeks 7-8: Bonus objectives 1-3

- 1-3 bonus objective tasks will be stack ranked, investigated and implemented.
 - o The number implemented will be based on the task difficulty.
- Begin collating reports and research so far into final report format.
- **Deliverable: Feature-improved app**
- **Deliverable: First final report attempt**
- **Supervisor review meeting & Next Timebox planning for bonus objectives/features**

Remaining weeks:

- Repeat of breakdown from weeks 7-8, iterating app features and report.
- **Deliverable: Further Feature-improved app**
- **Deliverable: Next final report attempt**
- **Supervisor review meeting & Next Timebox planning for bonus objectives/features**

This concludes my initial plan for the CM 3203 Project. I may augment / change this planning document as development progresses, and will note any significant alterations as part of my final report.

References:

[1] – About Apache Cordova – Published by the Apache Foundation, as seen on <http://cordova.apache.org/> at 19:25, Jan 30th 2015.

[2] – Xamarin Platform – published by Xamarin Inc, as seen on <http://xamarin.com/platform> at 19:29 Jan 30th 2015.

[3] – “This is Spartan, Microsoft’s new browser to challenge Google Chrome” – Published by the Next Web, as seen on <http://thenextweb.com/insider/2015/01/21/spartan-microsofts-new-browser-challenge-google-chrome/> at 20:30, Jan 31st 2015

[4] – “Material Design” – published by Google, as seen on <http://www.google.com/design/spec/material-design/introduction.html> at 21:00, Feb 1st 2015.