**News Miner (Content Aggregator Solution) Software Requirements Specification**

**VERSION 1.0**



**Group Id:** F1702E1067

**Supervisor Name :** Muhammad Saeed (muhammad.saeed@vu.edu.pk)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 25/11/2017 | 1.0 | Initial Draft | MC160401611 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# GIT Repo Location

https://github.com/furqanbaqai/F1702E1067.git

Table of Contents

[1 Scope of Project 3](#_Toc499405481)

[2 Functional and non-Functional Requirements 3](#_Toc499405482)

[2.1 Functional Requirements 3](#_Toc499405483)

[2.2 Non-Functional Requirements 4](#_Toc499405484)

[3 Use-Case Diagram 4](#_Toc499405485)

[4 Usage Scenarios 5](#_Toc499405486)

[4.1 Use-Case#1: Display News Content 5](#_Toc499405487)

[4.2 Use-Case#2: Search News Content 5](#_Toc499405488)

[5 Adopted Methodology 6](#_Toc499405489)

[5.1 Overview of the Agile Methodology 6](#_Toc499405490)

[5.2 Agile Methodology Application on the Project 6](#_Toc499405491)

[5.2.1 Content Repository 6](#_Toc499405492)

[5.2.2 Content Scrapping Module 6](#_Toc499405493)

[5.2.3 API Gateway 7](#_Toc499405494)

[5.2.4 Mobile App for Displaying News Content 7](#_Toc499405495)

[6 Work Plan 8](#_Toc499405496)

# Scope of Project

Scope of the solution is to develop a smart application which aggregates data from different English language news portal(s) in Pakistan and provides a heuristic view to the readers. The idea behind the solution is not to create OR change content, but to provide another view of the news to the readers by comparing, linking OR cross linking between different news providers.

Developed solution will consist of a mobile application (smart app) which will display news from different news portal. This solution will work in a server / client architecture where client will be a mobile application and server will be online services (based on restful services architecture).

Server component of the solution will work as a content repository and will save content scrapped from different news portal. Content Repository (Content Manager) will save content ranking, taxonomy of the content and content relation as well.

# Functional and non-Functional Requirements

## Functional Requirements

The underline idea behind this solution is to develop aggregated content repo of meta content so that it can be correlated with different other sources example Social Media (like Twitter, Facebook) and etc.

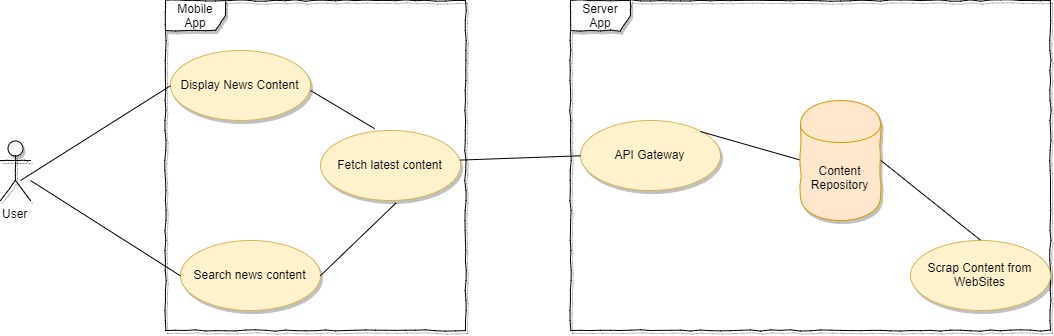
High level requirements form the solution are:

1. Solution will contain of two components:
   1. Client Component: Consisting of a mobile application (on android) used for displaying news to the reader.
   2. Server Component: Content Repository (Content Manager) having content store / repository of all news items. Client application will do a restful service call to the server and will fetch content from the repository.
2. Mobile application should support android devices on devices having varying screen resolutions.
3. Mobile application should be able to display news in following categories:
   1. Source: Example; dawn.com, geo.tv, tribune.com.pk (etc.)
   2. Content Placement: Example; heading, featured, front page, second page, etc.
4. Server application should have content delivery policy which should include (but not confined to):
   1. Content Expiry
   2. Content co-relation with same news on different sources

## Non-Functional Requirements

1. Solution should be robust, and servers should be able to pull out content within acceptable response time.
2. Client / server communication should be secure
3. Content repository should be properly indexed using criteria like content hash code, date time etc.

# Use-Case Diagram



# Usage Scenarios

## Use-Case#1: Display News Content

|  |  |
| --- | --- |
| **Objective** | Objective of this use-case is to display news content on the user’s mobile device |
| **Input Criteria** | Application is opened on the mobile device. |
| **Assumptions** | |
| 1. Specific mobile device has internet connection 2. Content is scrapped and saved into the repository 3. Back-end service interface is available | |
| **Execution Steps** | |
| 1. User open mobile application on his / her mobile 2. Mobile application sends an online request to the server to fetch required information 3. Mobile application loads all data and display news content as per the UI layout 4. In-case user click specific news content, detail news is displayed | |
| **Exit Criteria** | |
| All Content is displayed | |

## Use-Case#2: Search News Content

|  |  |
| --- | --- |
| **Objective** | Objective of this use-case is to search news content using free text search |
| **Input Criteria** | Application is opened on the mobile device. |
| **Assumptions** | |
| 1. Specific mobile device has internet connection 2. Content is scrapped and saved into the repository 3. Back-end service interface is available | |
| **Execution Steps** | |
| 1. User open mobile application on his / her mobile 2. Mobile application sends an online request to the server to fetch required information 3. User enters news content OR keywords in a free text field. 4. Application calls backend procedures to call online service and passes search criteria as a service parameter 5. Application display search response | |
| **Exit Criteria** | |
| All Content is displayed | |

## Use-Case#3: Fetch Content

|  |  |
| --- | --- |
| **Objective** | Objective of this use-case is to fetch latest content from content repository |
| **Input Criteria** | Application is opened on the mobile device. |
| **Assumptions** | |
| 1. Specific mobile device has internet connection 2. Content is scrapped and saved into the repository 3. Back-end service interface is available | |
| **Execution Steps** | |
| 1. Library establishes connection with backend servers 2. Specific request is send to the API gateway 3. API Gateway respond back with content’s meta content and summary as well. 4. Incase content is not received during specific time; response timeout error will be displayed | |
| **Exit Criteria** | |
| Response is received | |

## Use-Case#4: API gateway / Request Processing

|  |  |
| --- | --- |
| **Objective** | Objective of this use-case is to generate response of the API request |
| **Input Criteria** | Application is opened on the mobile device. |
| **Assumptions** | |
| 1. Application server having JSON API engine | |
| **Execution Steps** | |
| 1. API gateway authenticates the initiator. 2. Request is parsed, and specific backend procedure type is identified. 3. Response is intercepted from backend systems and is routed back to the caller | |
| **Exit Criteria** | |
| Response is generated and sent back to the calling application | |

## Use-Case#5: Scrap Content from News Portal

|  |  |
| --- | --- |
| **Objective** | Objective of this use-case is to scrap the content from news portal |
| **Input Criteria** | Backend script is triggered through cron job |
| **Assumptions** | |
| 1. There will be no interface of this daemon | |
| **Execution Steps** | |
| 1. Scraping application does HTTP-GET from specific news portal 2. HTML is parsed, and specific tags are scanned 3. Content is scrapped and saved to content repository | |
| **Exit Criteria** | |
| Content is saved into content repository | |

# Adopted Methodology[[1]](#footnote-1)

In-order to ensure maximum output and productivity, Agile development model will be used. This model will allow minimum solution development lifecycle and maximum output.

## Overview of the Agile Methodology

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like:

* Planning
* Requirements Analysis
* Design
* Coding
* Unit Testing and
* Acceptance Testing

Following are the Agile Manifesto principles:

* **Individuals and interactions** − In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
* **Working software** − Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
* **Customer collaboration** − As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
* **Responding to change** − Agile Development is focused on quick responses to change and continuous development.

## Agile Methodology Application on the Project

Agile will be applied by delivering following components in modular fashion:

### Content Repository

Whole solution is moves around content repository which will contain all news content in required taxonomical format. Development and integration of this content will be done first.

### Content Scrapping Module

A separate module will be developed which will scrap content from different websites and stores it in content repository within required taxonomy.

### API Gateway

API gateway will be developed / configured which will fetch all information from content repository and provide JSON based interface to the client.

### Mobile App for Displaying News Content

Smart, responsive mobile app will be developed to display content as per the requirements.

# Work Plan





1. Part of it’s content is extracted from [tutorialspoint.com](https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm) and from [agilemodeling.com](http://www.agilemodeling.com/essays/introductionToAM.htm) [↑](#footnote-ref-1)