VeriTag

Martin Maxim, Nick Bilotti, Jud Turner, Peter Halvorsen

Epic 1: User Interface and Navigation

User Stories:

1. As a user, I want to open the website and access a simple and intuitive homepage so that I can quickly understand the layout and find what I need.
2. As a user, I want a browsing section that lists headlines from multiple sources so that I can easily explore different articles.
3. As a user, I want to access and read entire articles in a clean, readable format so that I can focus on the content without distractions.
4. As a user, I want to see general comments and engage with discussions on articles so that I can participate in conversations and understand diverse perspectives.
5. As a programmer of VeriTag, I want to have manually loaded articles as a placeholder for the screen scraper so that I can best design the data in the UI.

Epic 2: Article Data Aggregation and Presentation

User Stories:

1. As a user, I want to see at least 5 articles from each of 3–4 sources, updated every 4 hours so that I have access to fresh and diverse content regularly.
2. As a user, I want to see the average total rating for each article on the Browse Articles page so that I can quickly gauge its overall quality.
3. As a user, I want to view detailed category ratings (bias, accuracy, quality) by clicking a pop-up button on the average rating icon so that I can better understand how the article was evaluated.

Epic 3: User Authentication and Interaction

User Stories:

1. As a verified user, I want to log in securely and see my status reflected on the website so that I can access additional features and trust my data is safe.

2. As a general user, I want the ability to post general comments on articles and vote on other users' comments so that I can participate in discussions and share my opinions.
3. As a verified user, I want an additional button to add my own verified comments on articles with ratings for each category and a brief explanation so that I can provide credible and constructive feedback.

Epic 4: User Feedback and Engagement

User Stories:

1. As a user, I want to like or dislike verified user comments so that I can signal their usefulness to other readers.
2. As a user, I want to upvote or downvote general user comments, with those receiving -5 votes automatically deleted so that the quality of discussions is maintained.
3. As a verified user, I want my comments to stand out as trustworthy and informative to other readers so that I can build credibility and add value to discussions.
4. As a user, I want to ensure that article feedback (comments, ratings) contributes to building a helpful reading experience so that the platform continuously improves and meets users' needs.

System Description:

This project is a web platform designed to aggregate articles from multiple sources, encourage meaningful discussions, and provide ratings to evaluate article quality and trustworthiness. It aims to create a better reading experience by integrating user feedback and verified insights.

Functional Requirements:

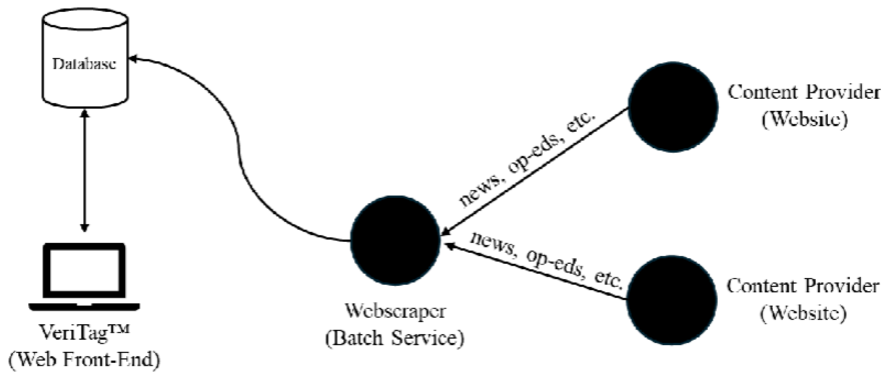
- Open the website
- Browsing section with headlines of articles screen scraped from 3 or 4 different sources
- Get at least 5 articles from every source, update every 4 hours
- On the Browse Articles page, show average total rating for each article
- Pop up button on average rating icon showing averages for each category (bias, accuracy, quality)
- Verified user login in the top right
- Open an article
- See verified user comments on the article with their ratings for each category and a brief explanation
- Can like or dislike the verified user comment
- See whole article

- General comments below
- Upvote / Downvote general user comments (-5 votes deleted)
- If you login as a verified user, basically the same but there is an extra button to add verified users comment

Non Functional Requirements:

- Non-Functional Req. 1 — System Response Time
  The system should respond to user interactions, such as navigating between pages or posting comments, within 2 seconds.
- Non-Functional Req. 2 — Data Freshness
  The system must update article data from all sources every 4 hours to ensure users see the latest headlines and ratings.
- Non-Functional Req. 3 — Availability
  The service must maintain an uptime of at least 99.5% over a 30-day period.
- Non-Functional Req. 4 — Security
  The service must support secure verified user logins by enforcing encrypted connections using HTTPS.
- Non-Functional Req. 5 — Scalability
  The system must handle up to 50 concurrent users without degradation in performance.
- Non-Functional Req. 6 — Legality
  The system must scrape legal content and not infringe on rights of other companies
- Non-Functional Req. 7 — Cordiality
  The system must encourage civil discourse and discourage chaotic and inappropriate dialogue

High-Level System Diagram:


Database
Content Provider (Website)
news, op-eds, etc.
news, op-eds, etc.
Content Provider (Website)
VeriTag™ (Web Front-End)
Webscraper (Batch Service)

Technologies Used

- Web Front End will be built in Pycharm and html website functionality
- Database will be made in Pycharm using SQLite schema
- Webscraper will be built in Pycharm and connect to the website front end

- Content providers will be free articles from reputable websites, perhaps transitioning to bigger sites that cost something in the future

Roadmap:

- Sprint 0 (2/6): Planning document finished
- Sprint 1 (2/20): Start Epic 1
- Sprint 2 (3/6): Finish Epic 1 and start Epic 3
- Sprint 3 (3/20): Finish epic 3
- Sprint 4 (4/3): Epic 4
- Sprint 5 (4/15): Start epic 2
- Sprint 6 (4/29): Finish epic 2, add anything else we want
- Due: May 2nd
- Present: May 5th