

Sprint Planning Meeting Minutes:

Attendees: Diego Martinez, Christian Cevallos, Jhonny Felix

Start time: <10:00pm>

End time: <10:30pm >

After discussion, the velocity of the team was estimated to be:

User Story <VER-MVP-002 CaptureArticleText> : 5 points

User Story <VER-MVP-003-SendArticleDataToBackend> : 5 points

User Story <VER-MVP-012-ManageUserSettings> : 3 points

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

4.2 Use Case 2 – Capture Article Text

Use Case ID: VER-MVP-002-CaptureArticleText
Level: Internal system process

User Story:
As the Veritas extension, I want to capture the visible article text and URL so that relevant content can be analyzed.

Actor:
Veritas Browser Extension

Pre-Conditions:

- Use Case 1 has started
- Content script is allowed to access the DOM

Trigger:
Analysis is requested by the user

System Behavior:

1. Content script identifies the article container
2. Extracts visible article text and removes non-article elements
3. Captures URL and metadata
4. Prepares data for backend submission

Post-Conditions:

- Cleaned article text and metadata are ready for transfer

Alternate Flow:

- Fallback extraction from the entire page body

Extensions:

- Site-specific extraction rules (future)

Exceptions:

- Empty or invalid extraction result

Concurrent Use:

- Can run independently across multiple tabs

Decision Support:

- Frequency: Once per analysis
- Criticality: High
- Risk: Medium

Constraints:

- Should not disrupt page layout
- Extraction must complete quickly

4.3 Use Case 3 – Send Article Data to Backend

Use Case ID: VER-MVP-003-SendArticleDataToBackend
Level: Internal system process

User Story:
As the Veritas extension, I want to send the captured article text and URL to FastAPI so that the article can be processed.

Actor:
Veritas Browser Extension

Pre-Conditions:

- Article text and URL are captured
- Backend endpoint is configured

Trigger:
Extension initiates the analysis request

System Behavior:

1. Forms a JSON payload with article data
2. Sends POST request to the FastAPI endpoint
3. Updates UI to processing state

Post-Conditions:

- Backend has received or attempted to receive the payload

Alternate Flow:

- Retry once if network issues occur

Extensions:

- Support for authentication headers in future

Exceptions:

- HTTP errors or request timeouts

Concurrent Use:

- Can be executed from multiple tabs concurrently

Decision Support:

- Frequency: Once per analysis
- Criticality: High
- Risk: Medium

Constraints:

- User must see a clear indication that analysis began
- Acceptable network latency must be maintained

4.12 Use Case 12 – Manage User-Friendly Settings

Use Case ID: VER-MVP-012-ManageUserSettings
Level: System-level UI interaction

User Story:
As a user, I want to configure Veritas settings through a clear and simple interface so that the tool behaves according to my preferences without requiring technical knowledge.

Actor:
User

Pre-Conditions:

- Veritas extension is installed and enabled
- Extension UI is accessible from the browser toolbar

Trigger:
User opens the settings panel from the Veritas extension interface.

System Behavior:

1. Extension displays a dedicated settings view
2. Current user preferences are loaded from local storage
3. Settings are grouped into clear categories such as display, analysis behavior, and performance
4. Each option includes descriptive labels and helper text
5. User modifies one or more settings using toggles, sliders, or dropdowns
6. Changes are validated in real time
7. Updated preferences are saved automatically

Post-Conditions:

- User preferences are persisted and applied to future analyses

Alternate Flow:

- User exits settings without making changes

Extensions:

- Preset profiles such as Minimal, Standard, and Detailed (future enhancement)

Exceptions:

- Local storage unavailable or write failure

Concurrent Use:

- Settings are scoped per browser profile

Decision Support:

- Frequency: Low to Medium
- Criticality: Medium
- Risk: Low

Constraints:

- Settings UI must be understandable without documentation
- No setting should require more than one interaction to enable or disable
- Changes must not require a browser refresh to take effect

The team members indicated their willingness to work on the following user stories.

- <Diego Martinez & Christian Cevallos>
User Story <VER-MVP-002 CaptureArticleText>
- <Diego Martinez & Christian Cevallos>
User Story <4.3 Use Case 3 – Send Article Data to Backend>

- <Jhonny Felix>
User Story <VER-MVP-012-ManageUserSettings>