

ユースケース名	新規登録
概要	ユーザーが新しいアカウントを作成する。
アクター	ユーザー
事前条件	サインアップページにアクセスできること。
事後条件	新しいユーザーアカウントが作成される。
基本フロー	<ol style="list-style-type: none">1. ユーザーはサインアップページにアクセスする。2. システムはサインアップフォームを表示する。3. ユーザーはユーザー名とパスワードを入力し、送信する。4. システムは入力内容を検証し、ユーザー名が既に存在しないことを確認する。5. システムは新しいアカウントを作成し、データベースに保存する。6. システムはユーザーに登録成功のメッセージを表示し、ログインページにリダイレクトする。
代替フロー	<p>4a. 入力内容が無効な場合</p> <ul style="list-style-type: none">4a1. システムはエラーメッセージを表示し、再入力を促す。4a2. ユーザーは修正後、再度送信する。

ユースケース名	ログイン
概要	ユーザーが既存のアカウントにログインする。
アクター	ユーザー
事前条件	ログインページにアクセスできること。
事後条件	ユーザーが認証され、メインページにアクセスできるようになる。
基本フロー	<ol style="list-style-type: none">1. ユーザーはログインページにアクセスする。2. システムはログインフォームを表示する。3. ユーザーはユーザー名とパスワードを入力し、送信する。4. システムはユーザーの存在とパスワードを検証する。5. システムはユーザーを認証し、メインページにリダイレクトする。
代替フロー	<p>4a. 認証が失敗した場合</p> <ul style="list-style-type: none">4a1. システムはエラーメッセージを表示し、再入力を促す。4a2. ユーザーは修正後、再度送信する。

ユースケース名	日記作成
概要	ユーザーが新しい日記エントリーを作成する。
アクター	ユーザー
事前条件	ユーザーがログインしていること。
事後条件	新しい日記エントリーが保存され、感情分析が行われる。
基本フロー	<ol style="list-style-type: none">1. ユーザーは日記作成ページにアクセスする。2. システムは日記作成フォームを表示する。3. ユーザーは日記の内容を入力し、送信する。4. システムは入力内容を検証する。5. システムは新しい日記エントリーを保存する。6. システムは感情分析を実行し、結果を保存する。7. システムはユーザーに日記作成成功のメッセージを表示する。
代替フロー	<ol style="list-style-type: none">4a. 日記の内容が無効な場合<ol style="list-style-type: none">4a1. システムはエラーメッセージを表示し、再入力を促す。4a2. ユーザーは修正後、再度送信する。

ユースケース名	日記編集
概要	ユーザーが既存の日記エントリーを編集する。
アクター	ユーザー
事前条件	ユーザーがログインしていること、編集する日記の所有者であること。
事後条件	日記エントリーが更新され、再度感情分析が行われる。
基本フロー	<ol style="list-style-type: none">1. ユーザーは編集したい日記エントリーの編集ページにアクセスする。2. システムは現在の日記内容を表示する。3. ユーザーは新しい内容を入力し、送信する。4. システムは入力内容を検証する。5. システムは日記エントリーを更新する。6. システムは再度感情分析を実行し、結果を保存する。7. システムはユーザーに日記編集成功のメッセージを表示する。
代替フロー	<p>4a. 日記の内容が無効な場合</p> <ul style="list-style-type: none">4a1. システムはエラーメッセージを表示し、再入力を促す。4a2. ユーザーは修正後、再度送信する。

ユースケース名	日記削除
概要	ユーザーが既存の日記エントリーを削除する。
アクター	ユーザー
事前条件	ユーザーがログインしていること、削除する日記の所有者であること。
事後条件	日記エントリーがデータベースから削除される。
基本フロー	<ol style="list-style-type: none">1. ユーザーは削除したい日記エントリーの削除ページにアクセスする。2. システムは削除確認画面を表示する。3. ユーザーが削除を確認する。4. システムは日記エントリーをデータベースから削除する。5. システムはユーザーに日記削除成功のメッセージを表示する。
代替フロー	<p>4a. 日記が見つからない場合</p> <p>4a1. システムはエラーメッセージを表示し、日記一覧ページにリダイレクトする。</p>

ユースケース名	日記閲覧
概要	ユーザーが自分の日記エントリーを閲覧する。
アクター	ユーザー
事前条件	ユーザーがログインしていること。
事後条件	ユーザーが自分の日記エントリーを閲覧できる。
基本フロー	1. ユーザーは日記一覧ページにアクセスする。 2. システムはユーザーの日記エントリーを取得する。 3. システムは日記エントリーのリストを表示する。
代替フロー	2a. ユーザーが日記を持っていない場合 2a1. システムは警告メッセージを表示する。

ユースケース名	チャットメッセージ送信
概要	ユーザーがチャットメッセージを送信する。
アクター	ユーザー
事前条件	ユーザーがログインしていること。
事後条件	新しいチャットメッセージがデータベースに保存され、AIからの応答が生成される。
基本フロー	<ol style="list-style-type: none">1. ユーザーはチャットページにアクセスする。2. システムは過去のチャットメッセージを表示する。3. ユーザーは新しいメッセージを入力し、送信する。4. システムはメッセージ内容を検証する。5. システムは新しいチャットメッセージを保存する。6. システムはAIによる応答を生成し、保存する。7. システムはユーザーに応答メッセージを表示する。
代替フロー	<p>4a. メッセージ内容が無効な場合</p> <ul style="list-style-type: none">4a1. システムはエラーメッセージを表示し、再入力を促す。4a2. ユーザーは修正後、再度送信する。

ユースケース名	チャット履歴閲覧
概要	ユーザーが自分のチャット履歴を閲覧する。
アクター	ユーザー
事前条件	ユーザーがログインしていること。
事後条件	ユーザーが自分のチャット履歴を閲覧できる。
基本フロー	1. ユーザーはチャットページにアクセスする。 2. システムはユーザーのチャット履歴を取得する。 3. システムはチャットメッセージのリストを表示する。
代替フロー	2a. ユーザーがチャット履歴を持っていない場合 2a1. システムは警告メッセージを表示する。

ユースケース名	チャット履歴削除
概要	ユーザーが自分のチャット履歴を削除する。
アクター	ユーザー
事前条件	ユーザーがログインしていること。
事後条件	チャット履歴がデータベースから削除される。
基本フロー	<ol style="list-style-type: none">1. ユーザーはチャット履歴削除ページにアクセスする。2. システムは削除確認画面を表示する。3. ユーザーが削除を確認する。4. システムはチャット履歴をデータベースから削除する。5. システムはユーザーに削除成功のメッセージを表示する。
代替フロー	<p>4a. チャット履歴が見つからない場合</p> <p>4a1. システムはエラーメッセージを表示し、チャットページにリダイレクトする。</p>

ユースケース名	感情分析結果確認
概要	ユーザーが自分の日記に対する感情分析結果を確認する。
アクター	ユーザー
事前条件	ユーザーがログインしており、日記エントリーが存在すること。
事後条件	感情分析結果が表示される。
基本フロー	1. ユーザーは感情分析結果ページにアクセスする。 2. システムはユーザーの日記エントリーを取得する。 3. システムは感情分析結果を生成する。 4. システムは感情分析のグラフと円グラフを表示する。
代替フロー	2a. ユーザーが日記を持っていない場合 2a1. システムは警告メッセージを表示する。