Use Case: User Signup

Actor: Visitor
Goal: Register a new account and verify it to access the full platform
Trigger: User submits the signup form

Preconditions:

- Username and email must be unique
- Strong password policy enforced
- Account type selection required

Main Scenario:

1. Visitor accesses the /signup route
2. Fills in username, email, password, confirm password, and selects account type (SaaS, B2B, eCommerce, business, individual)
3. System validates all fields with frontend and backend checks
4. Password is hashed and stored securely
5. System generates email verification token
6. Sends confirmation email
7. User sees “Check your email to activate account”
8. Upon clicking the email link, user is marked as verified and redirected to login

Postcondition:

- Account is created, verified, and saved in DB
- Login access granted after email verification

Alternate Paths:

- Username/email exists → Form error
- Password too weak → Validation error
- Email not verified → Cannot login
- Email delivery fails → Notify admin/log error

Use Case: User Login

Actor: Registered and Verified User
Goal: Authenticate and begin session
Trigger: Login form submitted

Preconditions:

- User must be verified
- Must enter valid credentials

Main Scenario:

1. User inputs email/username, password, and account type
2. Backend validates input format and checks credentials
3. Checks account verification status
4. If 2FA enabled, prompt user for code
5. Session cookie or JWT token is issued
6. User is redirected to /dashboard

Postcondition: Authenticated session active

Alternate Paths:

- Wrong password → Error flash
- Unverified account → Resend verification link
- Account locked after 5 attempts → Cooldown or CAPTCHA

Use Case: Upload CSV and Predict Churn

Actor: Logged-in User
Goal: Upload custom dataset and receive intelligent churn insights

Trigger: Button click on "Upload and Predict"

Preconditions:

- CSV file must match accepted structure or allow mapping
- User must be authenticated

Main Scenario:

1. User selects and uploads CSV
2. System performs file validation, checks encoding, headers, size
3. Offers smart column mapping if headers don't match expected schema
4. Model is trained on uploaded dataset (Option B flow)
5. Predictions are made in real-time
6. Churn results are shown with visualizations (charts, metrics, risk segmentation)
7. Option to download results as enhanced report (PDF/CSV)

Postcondition:

- Predictions displayed and optionally saved in user session

Alternate Paths:

- Missing file → Alert
- Model training error → Retry suggestion
- Column mismatch → Mapping UI triggered
- File too large → Limit warning

Use Case: Predict Single Customer (API)

Actor: Authenticated Developer or App Integration
Goal: Receive churn probability via API

Trigger: POST request to /predict-single

Preconditions:

- Auth token or session required
- JSON must contain valid, clean schema

Main Scenario:

1. API receives payload
2. System authenticates token
3. Input JSON is validated and encoded
4. Trained model is loaded from server
5. Model returns churn prediction + probability
6. System classifies prediction into LOW/MEDIUM/HIGH risk
7. Returns response with prediction, confidence score, and metadata

Postcondition: Valid JSON response with detailed churn info

Alternate Paths:

- Token expired → 401 Unauthorized
- Malformed JSON → 400 Bad Request
- Missing fields → Schema validation error
- Model load error → 503 Service Unavailable

Use Case: View Dashboard

Actor: Logged-in User
Goal: Access personalized insights and past activity

Trigger: Navigating to /dashboard

Preconditions:

- Valid authenticated session

Main Scenario:

1. User logs in
2. System fetches previous uploads, plan status, user metadata
3. Displays churn predictions, recent activity, plan renewal info
4. Sidebar gives access to tools, settings, and file history

Postcondition: Dashboard with analytics and user specific context shown

Alternate Paths:

- Expired session → Redirect to login
- No previous data → show onboarding tutorial