**Requirements Specifications**

**P09:InterviewPrepApp**

| **Student ID** | **Name** |
| --- | --- |
| **26100314** | **Abdul Samad** |
| **26100049** | **Abdul Rehman** |
| **26100401** | **Muhammad kabli** |
| **26100251** | **Ahmed bilal** |
| **26100032** | **Haider Abbas Virk** |

**Table of Contents**

[1.](#_heading=h.aahtgg3y50j8) Introduction 3

[2.](#_heading=h.vc1zfci4zgr0) System Actors 4

[3.](#_heading=h.jmksjkfym2c4) Use Cases 5

[3.1 Use Case Diagrams 5](#_heading=h.8ktw7xuzgyzc)

[3.2 Description of Use Cases 5](#_heading=h.tnbf4z98ynfo)

[3.2.1 Withdraw cash 5](#_heading=h.7td8w8sab47a)

[3.2.2 Transfer funds 6](#_heading=h.kzau99a5oj00)

[4.](#_heading=h.8h60dp47zocs) Class Diagram 7

[4.1 Diagram 7](#_heading=h.rht8dq48kfc9)

[4.2 Description 7](#_heading=h.3vnneqtu78h3)

[5.](#_heading=h.df4jjy6blggu) Sequence Diagrams 8

[5.1 Use case Name e.g., Withdraw cash 8](#_heading=h.hilgr3v0i9pf)

[5.2 Use case Name e.g., Transfer funds 8](#_heading=h.9qzz3fqtpz1s)

[6.](#_heading=h.1pb84pfx7bn9) State Diagrams 9

[6.1 Diagram details 9](#_heading=h.mg4s0k5cy02q)

[6.2 Diagram 9](#_heading=h.gx6uzyiekvv2)

[7.](#_heading=h.95bwc8epwuf) Data Requirements 10

[8.](#_heading=h.j8m551xywbzb) Non-functional Requirements / Quality Attributes 11

[9.](#_heading=h.bqgoo9qiesj7) Security Requirements 12

[10.](#_heading=h.q04xk08346eu) Security Engineer 13

[11.](#_heading=h.dswsabcz02fi) Use of Generative AI 14

[12.](#_heading=h.nw38noq9ya5p) Who Did What? 15

[13.](#_heading=h.2wwnbkuf8pwj) Review checklist 15

# Introduction

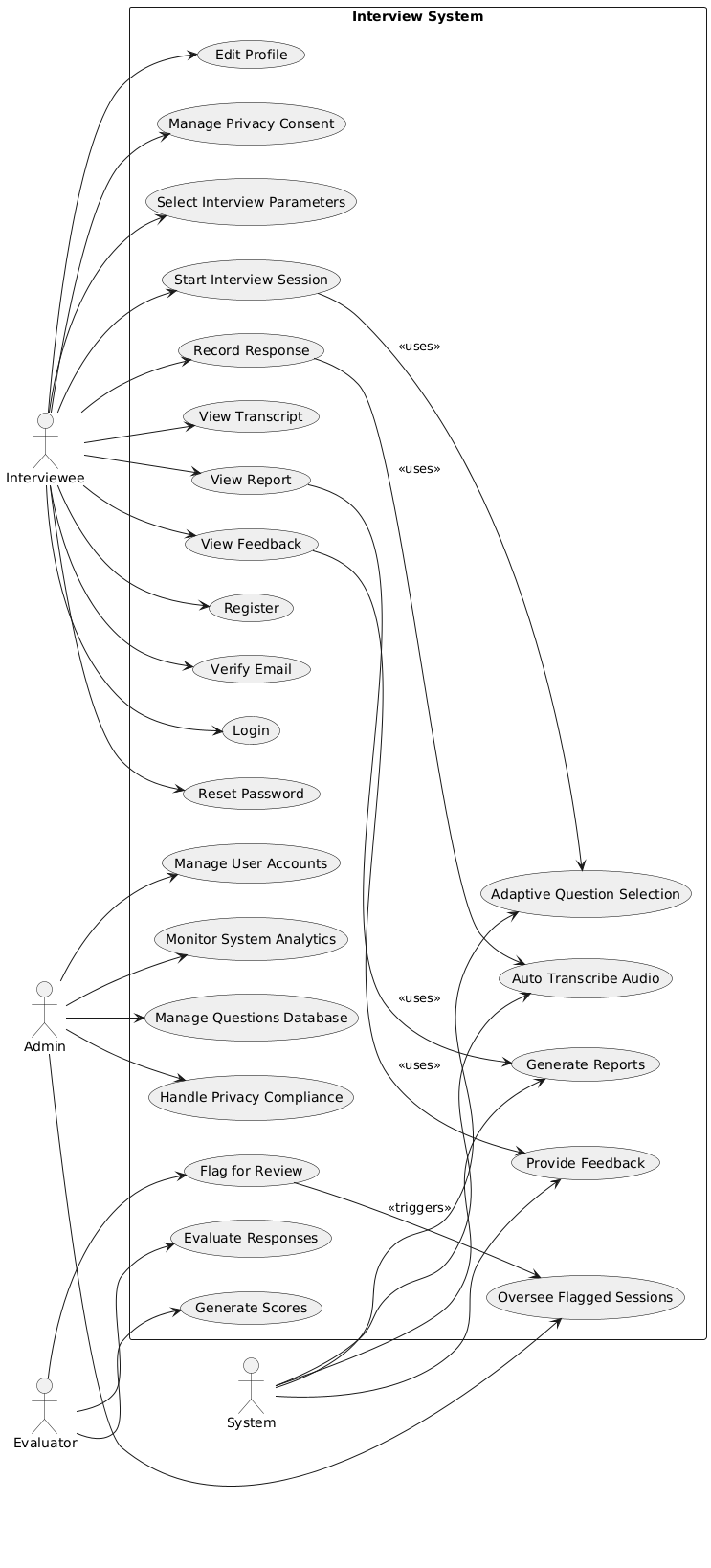
This project is an AI-powered interview preparation platform that allows job seekers to simulate real interview scenarios. The system collects job-related data from the user, generates role-specific questions using a large language model, and conducts a mock interview through conversational AI (e.g., Vapi or ElevenLabs assistant). After the session, the system provides feedback on user performance, covering aspects such as communication clarity, confidence, technical correctness, and behavioral structure (e.g., STAR method).  
  
**Objectives:**  
- Help students and professionals prepare effectively for job interviews.  
- Provide personalized, role-specific, and skill-specific mock interviews.  
- Deliver actionable feedback to improve communication and technical skills.  
  
**Potential Users:**  
- University students preparing for placements.  
- Professionals switching jobs or industries.  
- Career counseling and training centers.

# System Actors

| **Actor Name** | **Description** |
| --- | --- |
| Interviewee (User) | The primary user of the app who practices interviews by answering questions via voice/video and receives feedback on their performance. |
| Admin | Manages user accounts, moderates peer-to-peer interactions and ensures the smooth operation and security of the platform. |
| Evaluator | A human interview expert who reviews the AI-generated interview reports and transcriptions of candidate responses. They provide personalized qualitative feedback, highlight areas for improvement (such as communication style, technical depth, or confidence), and may adjust scores if needed. This role complements the AI Interviewer by adding human judgment and ensuring more realistic and credible interview preparation for the user |

# Use Cases

## Use Case Diagrams



## Description of Use Cases

3.2.1

| **Identifier** | | UC-001: Register |
| --- | --- | --- |
| **Purpose** | | The Interviewee creates a new account in the system. |
| **Pre-conditions** | | The user has internet access and a valid email address. |
| **Post-conditions** | | A new account is created and stored; verification email is sent. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | The interviewee opens the app/website and selects 'Register'. | |
|  | The system displays the registration form (name, email, password). | |
|  | The interviewee fills the form and submits. | |
|  | The system validates inputs and creates a provisional account. | |
|  | The system sends a verification email to the provided address. | |
|  | The system displays confirmation that registration is successful pending verification. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee cancels registration before submitting. | |
| **Step #** | **Exception Paths** | |
|  | Email already in use; system displays an error and aborts registration. | |

### 

3.2.2

| **Identifier** | | UC-002: 2-Step Verification (Email) |
| --- | --- | --- |
| **Purpose** | | Verify the Interviewee's email address to activate the account. |
| **Pre-conditions** | | The interviewee has completed registration and has access to the registered email. |
| **Post-conditions** | | The account is verified and activation status is set to confirm. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | System sends a verification email containing a link or code. | |
|  | Interviewee opens the email and clicks the link or copies the code. | |
|  | Interviewee submits the code or follows the link in the app/website. | |
|  | System validates the code or token and marks the account as verified. | |
|  | System displays a confirmation message and allows login. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee requests a resend of the verification email. | |
| **Step #** | **Exception Paths** | |
|  | Verification link/code expired or invalid; system prompts to resend. | |

### 

3.2.3

| **Identifier** | | UC-003: Login |
| --- | --- | --- |
| **Purpose** | | Allow a registered and verified Interviewee to authenticate and access their dashboard. |
| **Pre-conditions** | | Interviewee has a registered and verified account. |
| **Post-conditions** | | Interviewee is authenticated and granted a session token; dashboard is displayed. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Interviewee navigates to the login page and selects 'Login'. | |
|  | Interviewee enters email and password. | |
|  | System validates credentials. | |
|  | If 2FA is enabled, system requests second factor and validates it. | |
|  | Upon success, system creates a session and redirects the Interviewee to the dashboard. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee cancels login or opts for SSO if available. | |
| **Step #** | **Exception Paths** | |
|  | Incorrect credentials: system displays an error; repeated failures may lock the account. | |

### 

3.2.4

| **Identifier** | | UC-004: Forgot / Reset Password |
| --- | --- | --- |
| **Purpose** | | Allow Interviewee to regain access by securely resetting their password. |
| **Pre-conditions** | | The interviewee has a registered email address. |
| **Post-conditions** | | Interviewee's password is updated and they can log in with the new password. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Interviewee clicks 'Forgot Password' on the login page. | |
|  | Interviewee enters the registered email address. | |
|  | System sends a password-reset link to the email. | |
|  | Interviewee clicks the link, which opens a secure reset form | |
|  | Interviewee enters a new password and submits. | |
|  | System updates the password and confirms success. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee cancels the password reset process | |
| **Step #** | **Exception Paths** | |
|  | Reset link expired or invalid; system prompts to request a new link. | |

### 

3.2.5

| **Identifier** | | UC-005: Edit Profile |
| --- | --- | --- |
| **Purpose** | | Interviewee updates personal profile and preferences. |
| **Pre-conditions** | | Interviewee is authenticated and on their profile page. |
| **Post-conditions** | | Profile changes are saved and reflected in the interview experience. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Interviewee opens the profile/settings page. | |
|  | Interviewee edits fields (name, bio, preferred role, languages). | |
|  | Interviewee updates preferences (time zone, notification preferences). | |
|  | Interviewee clicks 'Save' to submit changes. | |
|  | System validates and stores the updated profile information. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee cancels edits and discards changes. | |
| **Step #** | **Exception Paths** | |
|  | Invalid input (e.g., unsupported characters) prevents saving; system shows validation errors. | |

### 

3.2.6

| **Identifier** | | UC-006: Settings & Privacy / Consent |
| --- | --- | --- |
| **Purpose** | | Interviewee manages privacy settings and data consents. |
| **Pre-conditions** | | Interviewee is authenticated and navigates to settings. |
| **Post-conditions** | | Selected privacy and consent settings are saved and enforced. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Interviewee opens 'Settings' and selects 'Privacy & Consent'. | |
|  | The system displays options (data sharing, audio retention, transcripts). | |
|  | Interviewee toggles or selects preferences and consent options. | |
|  | Interviewee clicks 'Save' to apply choices. | |
|  | The system stores preferences and updates how data is handled. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee reverts to default privacy settings. | |
| **Step #** | **Exception Paths** | |
|  | Backend error prevents saving; system alerts the Interviewee. | |

### 

3.2.7

| **Identifier** | | UC-007: Select Interview Parameters |
| --- | --- | --- |
| **Purpose** | | Interviewee chooses role, company style, domain and difficulty bias for the session. |
| **Pre-conditions** | | The interviewee is authenticated and about to start an interview. |
| **Post-conditions** | | Selected parameters are stored and used to tailor question selection. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Interviewee clicks 'Start Interview' or 'Create Session'. | |
|  | System prompts for parameters: role, company profile, domain (algorithms, system design), format (behavioral, coding), difficulty preference and time limit. | |
|  | Interviewee selects values and confirms. | |
|  | The system saves parameters and prepares the session. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee skips selection to use default parameters. | |
| **Step #** | **Exception Paths** | |
|  | Invalid parameter combination flagged; system requests correction. | |

### 

3.2.8

| **Identifier** | | UC-008: Start Interview Session |
| --- | --- | --- |
| **Purpose** | | Initialize and begin an interview session with the AI Interviewer. |
| **Pre-conditions** | | Interview parameters are selected or defaulted; Interviewee is authenticated. |
| **Post-conditions** | | Interview session becomes active; session ID and resources allocated. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Interviewee clicks 'Start'. | |
|  | The system creates a new session record and allocates resources. | |
|  | The AI Interviewer greets the Interviewee and explains rules (timing, hints). | |
|  | The system presents the first question according to selected parameters. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | The interviewee cancels before answering any question. | |
| **Step #** | **Exception Paths** | |
|  | System resources unavailable or error creating a session; system notifies the Interviewee. | |

### 

3.2.9

| **Identifier** | | UC-009: Record User Response |
| --- | --- | --- |
| **Purpose** | | Capture the Interviewee's spoken or video answer for evaluation. |
| **Pre-conditions** | | The interview session is active and a question has been presented. |
| **Post-conditions** | | Recording is saved and associated with the session for later transcription and scoring. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | The system displays a 'Record' button and optional guidelines (time limit, allowed retries). | |
|  | The interviewee grants microphone/camera permission and starts recording. | |
|  | The interviewee speaks/records their answer within the allotted time. | |
|  | The interviewee stops the recording and confirms submission. | |
|  | The system saves the audio/video file to the session storage. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee aborts and retries recording before final submission. | |
| **Step #** | **Exception Paths** | |
|  | Microphone/camera permission denied or hardware error; recording cannot proceed. | |

### 

3.2.10

| **Identifier** | | UC-010: Transcribe Audio |
| --- | --- | --- |
| **Purpose** | | Convert recorded audio to text so evaluation and reviewer access are possible. |
| **Pre-conditions** | | A valid recording is stored for the current session. |
| **Post-conditions** | | Transcript is produced, stored, and attached to the session record. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | The system sends the audio file to an internal/external transcription service. | |
|  | Transcription service processes audio and returns a transcript. | |
|  | The system stores the transcript and marks the transcription as available. | |
|  | Interviewee or reviewer can view and edit the transcript if needed. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Interviewee manually corrects transcription errors. | |
| **Step #** | **Exception Paths** | |
|  | Transcription service fails or returns poor-quality text; system flags for manual review. | |

### 

3.2.11

| **Identifier** | | UC-011: Manage User Accounts |
| --- | --- | --- |
| **Purpose** | | Admin manages user accounts (suspend, reactivate, delete) and views account details. |
| **Pre-conditions** | | Admin is authenticated with account management privileges. |
| **Post-conditions** | | User account status is updated and logged in the audit trail. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Admin logs into the admin console. | |
|  | Admin navigates to the 'User Management' section. | |
|  | Admin searches for the target Interviewee account by email or ID. | |
|  | Admin selects an action (suspend, reactivate, delete) and confirms. | |
|  | The system applies the action and records it in the audit log. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Admin assigns a temporary suspension or warns the user instead of deletion. | |
| **Step #** | **Exception Paths** | |
|  | Target account not found or Admin lacks privileges; system displays an error. | |

### 

3.2.12

| **Identifier** | | UC-012: Monitor System Analytics |
| --- | --- | --- |
| **Purpose** | | Admin reviews platform metrics and health (usage, errors, sessions). |
| **Pre-conditions** | | Admin is authenticated and has analytics access |
| **Post-conditions** | | Admin views and can export analytics for decision making. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Admin opens the analytics dashboard from the admin console. | |
|  | The system displays key metrics (active users, sessions/day, flagged items, average scores). | |
|  | Admin applies filters (date range, segment) and inspects charts and tables. | |
|  | Admin exports or schedules reports as needed | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Admin adjusts dashboard widgets or shares a view with other admins. | |
| **Step #** | **Exception Paths** | |
|  | The analytics backend is temporarily unavailable; the system shows an error. | |

### 

3.2.13

| **Identifier** | | UC-013: Manage Content / Questions Database |
| --- | --- | --- |
| **Purpose** | | Admin adds, updates, or removes questions and maintains metadata (difficulty, tags). |
| **Pre-conditions** | | Admin is authenticated with content-editing rights. |
| **Post-conditions** | | The question bank is updated and changes are versioned/ audited. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Admin opens the Question Bank management UI. | |
|  | Admin selects 'Add Question' or chooses an existing question to edit. | |
|  | Admin fills/edit fields (prompt, expected outcomes, difficulty, tags, sample answers). | |
|  | Admin saves changes and optionally publishes to the active question pool. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Admin uploads a bulk import file to add many questions at once. | |
| **Step #** | **Exception Paths** | |
|  | Validation errors (missing fields) prevent saving; system highlights required fields. | |

### 

3.2.14

| **Identifier** | | UC-014: Handle Privacy and Data Compliance |
| --- | --- | --- |
| **Purpose** | | Admin processes user data requests (export/delete) to meet regulations. |
| **Pre-conditions** | | The user or system has submitted a privacy request (export/delete). |
| **Post-conditions** | | Requested data is exported or deleted and request status is updated. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Admin opens the 'Privacy Requests' panel in the admin console. | |
|  | Admin reviews the pending request details and verifies requester identity. | |
|  | Admin instructs the system to export or delete the specified data. | |
|  | System performs the operation and returns confirmation. | |
|  | Admin verifies completion and notifies the user. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Admin escalates a complex request to the legal team. | |
| **Step #** | **Exception Paths** | |
|  | Operation fails due to data retention policy or legal hold; system logs the failure. | |

### 

3.2.15

| **Identifier** | | UC-015: Oversee Flagged Sessions |
| --- | --- | --- |
| **Purpose** | | Admin reviews sessions flagged by AI or users for content or scoring issues. |
| **Pre-conditions** | | One or more sessions are flagged in the system. |
| **Post-conditions** | | Flagged sessions are reviewed and resolved (dismissed, corrected, or escalated). |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Admin opens the 'Flagged Sessions' queue in the admin console. | |
|  | Admin selects a flagged session and reviews transcript, recording and AI notes. | |
|  | Admin decides to dismiss the flag, correct AI outputs, delete content, or assign to an Evaluator. | |
|  | System records the action and updates the flag status. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Admin assigns flagged sessions in bulk to Evaluators for review. | |
| **Step #** | **Exception Paths** | |
|  | Flagged session data is corrupted or missing; system returns an error. | |

### 

3.2.16

| **Identifier** | | UC-016: Access Interview Reports |
| --- | --- | --- |
| **Purpose** | | Evaluator retrieves AI-generated interview reports and transcripts for human review. |
| **Pre-conditions** | | Evaluator is authenticated with reviewer privileges. |
| **Post-conditions** | | The evaluator can read the report, access recordings and transcripts, and prepare feedback. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Evaluator logs into the reviewer dashboard. | |
|  | Evaluator filters reports by criteria (date, candidate, score range). | |
|  | Evaluator selects a report to open. | |
|  | System displays the AI report, transcript, and recording links. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Evaluator requests raw audio or coding artifacts if needed. | |
| **Step #** | **Exception Paths** | |
|  | Report not available due to processing delays; system shows the status. | |

### 

3.2.17

| **Identifier** | | UC-017: Provide Qualitative Feedback |
| --- | --- | --- |
| **Purpose** | | Evaluator gives human-written, personalized feedback on a candidate's performance. |
| **Pre-conditions** | | The evaluator has opened a candidate's report and transcript. |
| **Post-conditions** | | Feedback is stored and attached to the candidate's report and visible to the Interviewee. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Evaluator reviews the AI-generated report and transcript. | |
|  | The evaluator writes qualitative comments, suggestions, and improvement tips. | |
|  | Evaluator tags strengths and weaknesses and may attach annotated transcript snippets. | |
|  | Evaluator submits feedback and chooses whether to notify the Interviewee. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Evaluator saves feedback as draft to continue later. | |
| **Step #** | **Exception Paths** | |
|  | Saving feedback fails due to a backend error; system prompts to retry. | |

### 

3.2.18

| **Identifier** | | UC-018: Adjust AI Scores |
| --- | --- | --- |
| **Purpose** | | Evaluator modifies AI-assigned scores when they are deemed inaccurate. |
| **Pre-conditions** | | The evaluator has reviewed the report and identified a scoring discrepancy. |
| **Post-conditions** | | Updated scores replace provisional AI scores and changes are logged with justification. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Evaluator reviews specific AI-scored metrics (correctness, communication, depth). | |
|  | Evaluator edits score values and provides a justification note. | |
|  | Evaluator submits adjustments. | |
|  | System updates the report, logs the change, and timestamps the adjustment. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Evaluator reverts to AI score if unsure. | |
| **Step #** | **Exception Paths** | |
|  | Scoring schema prevents edits for locked or audited sessions. | |

### 

3.2.19

| **Identifier** | | UC-019: Certify Candidate Progress |
| --- | --- | --- |
| **Purpose** | | Evaluator certifies that a candidate has achieved a readiness milestone. |
| **Pre-conditions** | | The evaluator has reviewed sufficient session history for the candidate. |
| **Post-conditions** | | Certification is added to the candidate profile with notes and validity period. |
|  | | |
| **Step #** | **Typical Course of Action** | |
|  | Evaluator opens the candidate's profile and performance history. | |
| **2.** | Evaluator selects 'Certify Progress' and chooses a certification level. | |
| **3.** | Evaluator enters supporting comments and the certification validity term. | |
| **4.** | Evaluator confirms and system records the certification on the profile. | |
|  | | |
| **Step #** | **Alternate Courses of Action** | |
|  | Evaluator defers certification pending further evidence. | |
| **Step #** | **Exception Paths** | |
|  | Insufficient evidence or conflicting data prevents certification. | |

### 

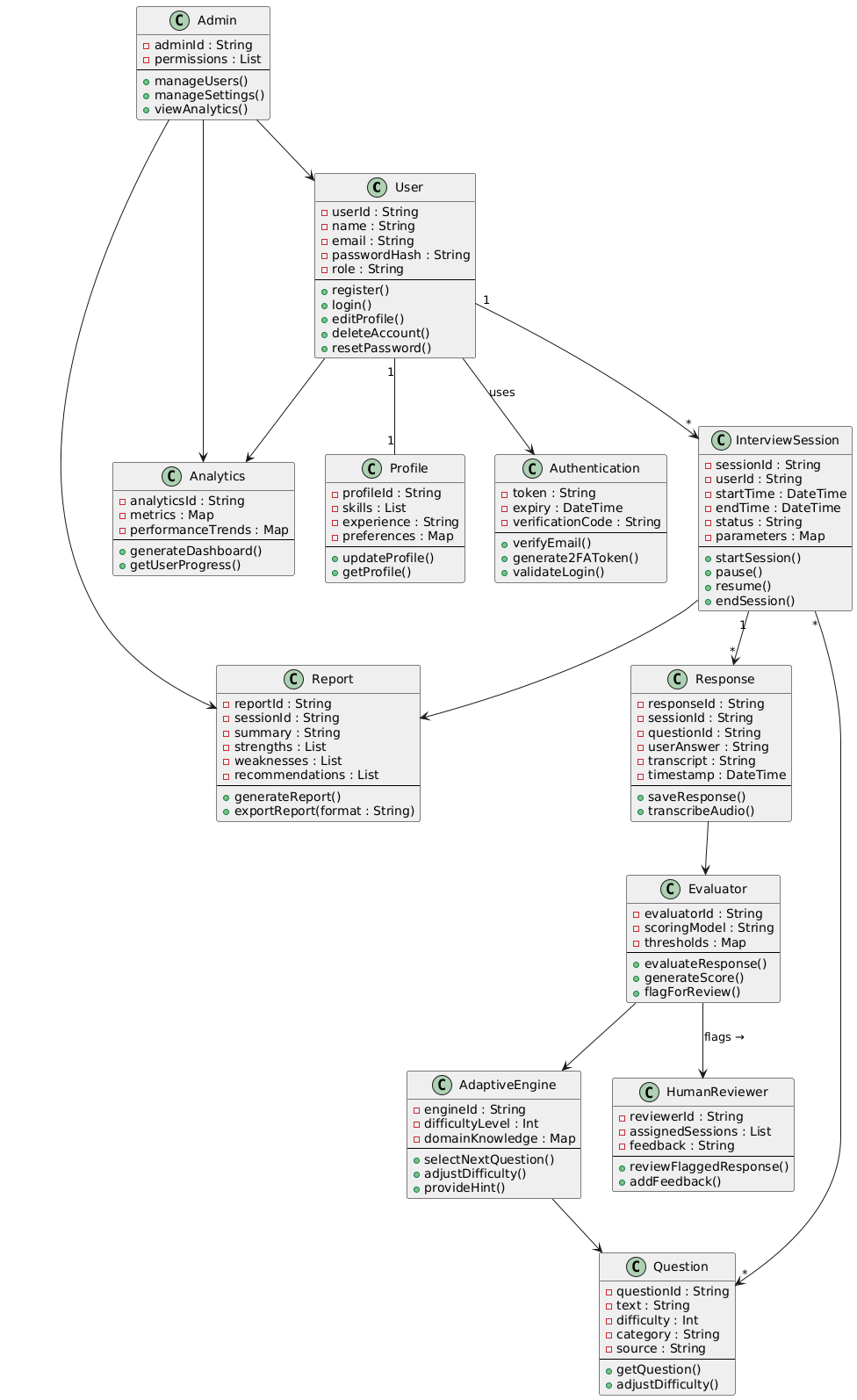
3.2.20

| **Identifier** | | UC-020: Notify Candidate of Human Review |
| --- | --- | --- |
| **Purpose** | | Inform Interviewee that a human Evaluator has reviewed their session and feedback is available. |
| **Pre-conditions** | | The evaluator has submitted feedback for a candidate's session. |
| **Post-conditions** | | Interviewee receives a notification and can view the evaluator's feedback in their dashboard. |
|  | | |
|  | **Typical Course of Action** | |
| **1.** | After submitting feedback, Evaluator selects 'Notify Candidate' or system auto-notifies based on settings. | |
| **2.** | The system creates a notification entry and queues an email (if enabled). | |
| **3.** | Interviewee receives in-app notification (and email if configured). | |
| **4.** | Interviewee views the feedback in the session report. | |
|  | | |
|  | **Alternate Courses of Action** | |
|  | The evaluator chooses not to notify immediately and schedules notification later. | |
|  | **Exception Paths** | |
|  | Notification delivery fails; system logs it and retries based on queue rules. | |

### 

# Class Diagram

## Diagram



## Description

### **User Management**

**User**Represents someone using the system. Keeps basic info like name, email, password hash, and role. Main actions are creating an account, logging in, editing details, deleting, or resetting the password.

**Profile**Holds extra details about the user such as skills, experience, and preferences. It can be updated or fetched when needed.

**Authentication**Handles login security. Manages tokens, expiry times, and verification codes. Used for email verification, 2FA tokens, and validating logins.

### **Interview Domain**

**InterviewSession**Tracks a single interview attempt. Stores timing, status (running/paused/ended), and chosen parameters. Lets the user start, pause, resume, or end their session.

**Question**Represents one interview question with its text, difficulty, category, and source. Can be fetched and difficulty adjusted.

**Response**Stores the user’s answer to a question, along with transcript and timestamp. Can save answers and convert audio to text.

**Evaluator**The scoring logic. Uses a model and thresholds to grade answers, generate scores, and flag anything unusual for review.

**AdaptiveEngine**Adjusts the flow of the interview. Chooses the next question, changes difficulty, and can provide hints based on performance.

### **Reports & Analytics**

**Report**Summarizes performance for one session. Contains strengths, weaknesses, and recommendations. Can be generated and exported.

**Analytics**Look across multiple sessions. Tracks metrics and trends, builds dashboards, and shows how a user is progressing.

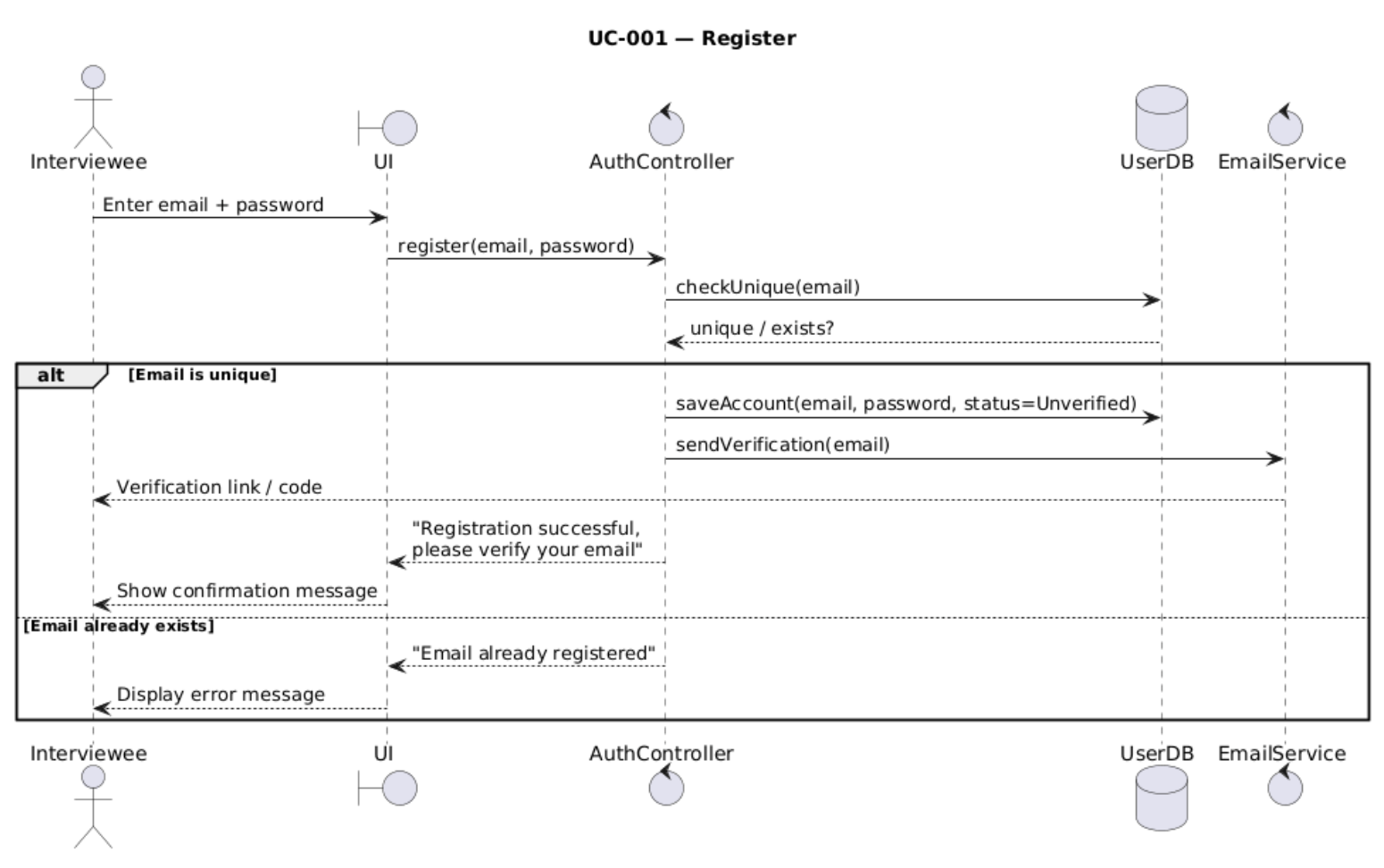
### **Review & Admin**

**HumanReviewer**A person who double-checks flagged responses. Can review answers and add feedback.

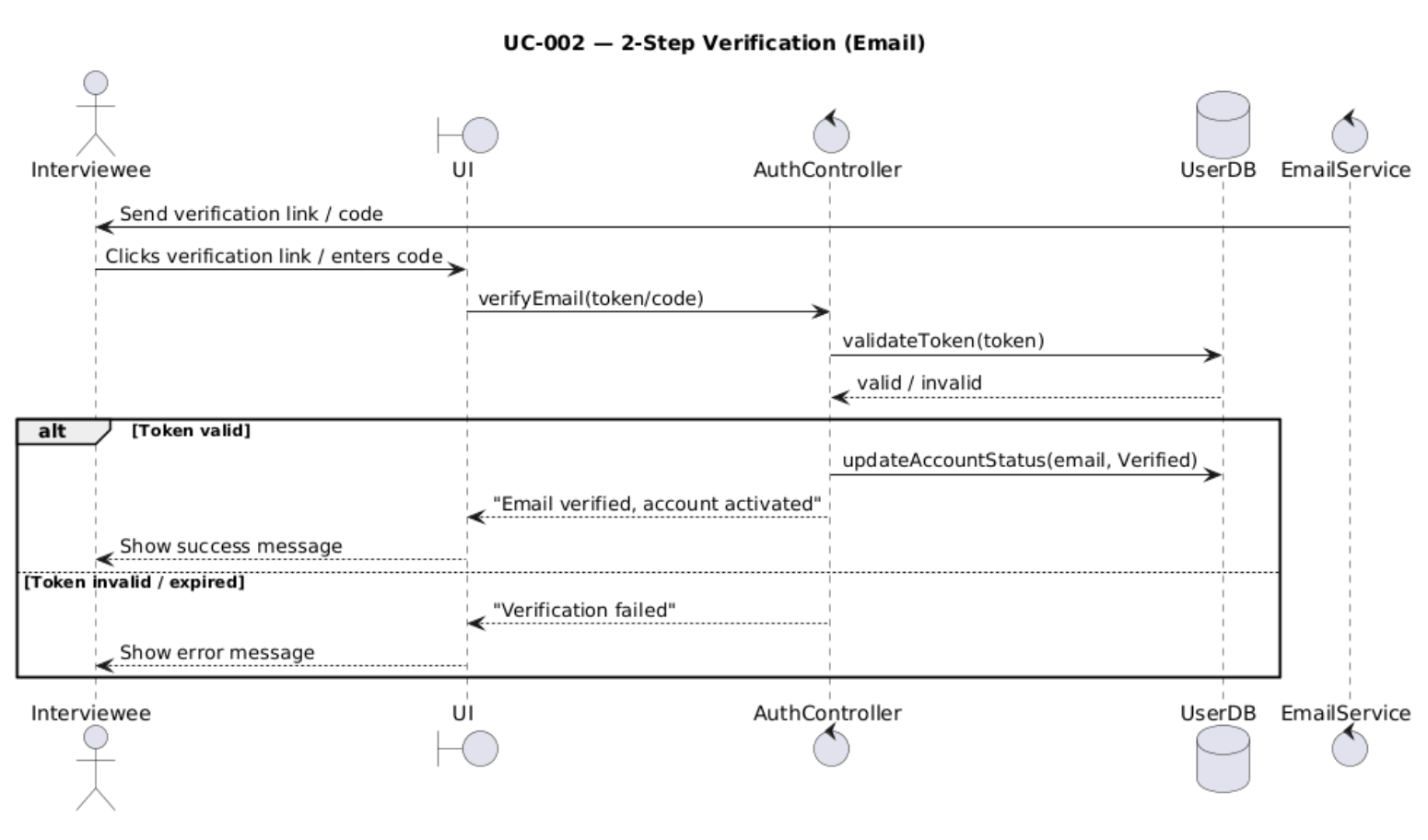
**Admin**Has system-level control. Can manage users, configure settings, and view overall reports and analytics.

# Sequence Diagrams

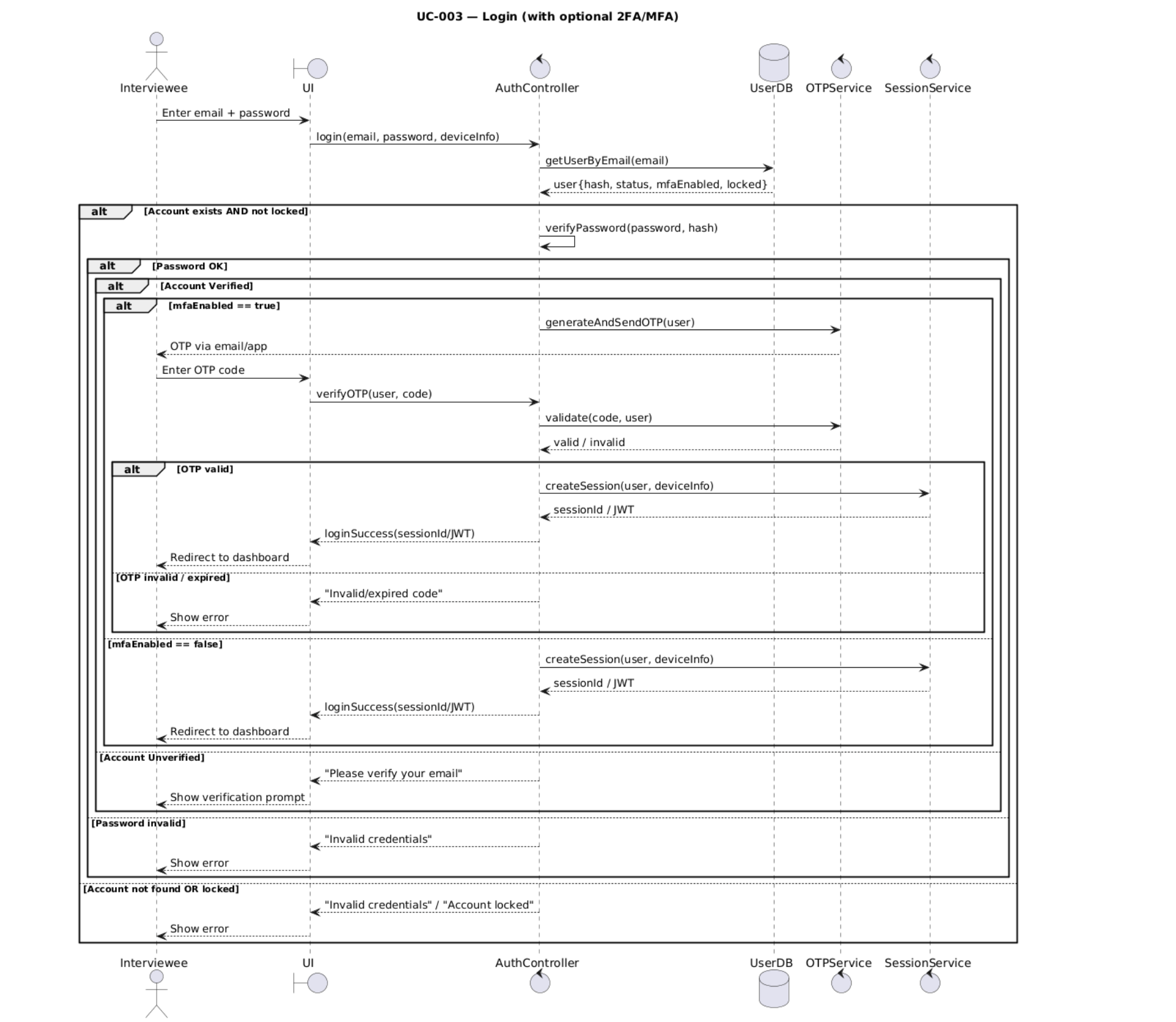
## Register



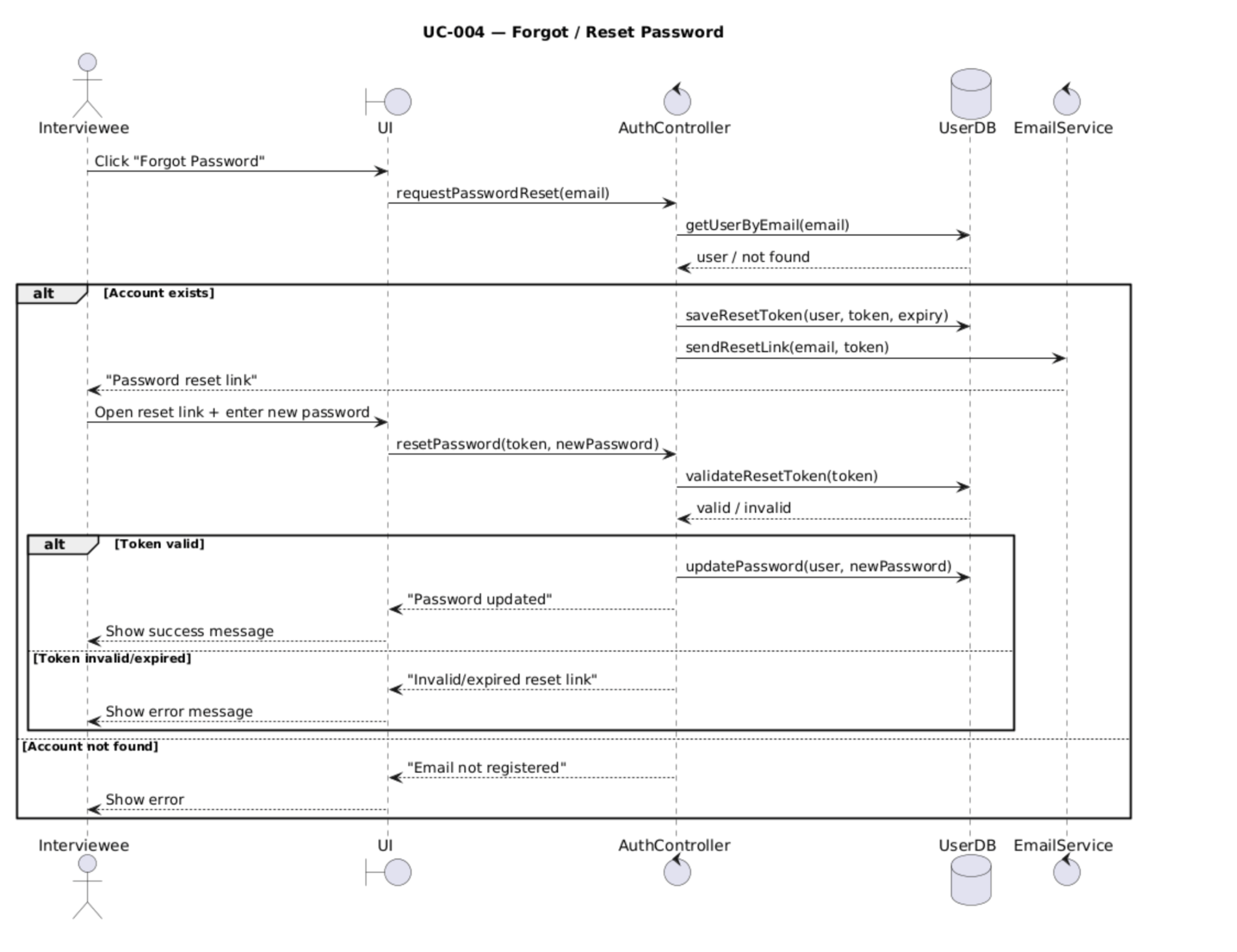
## 2-Step Verification (Email)



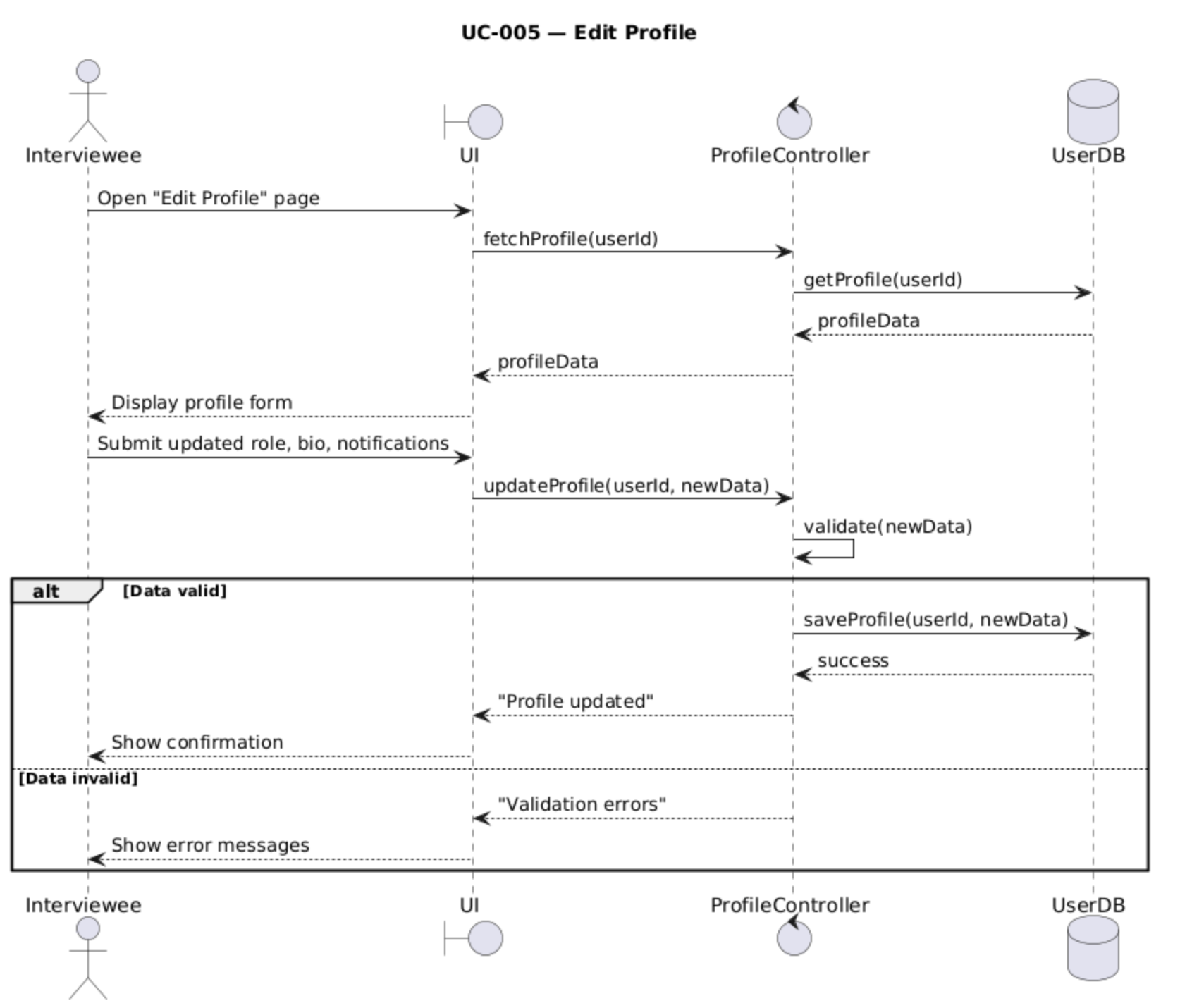
## Login



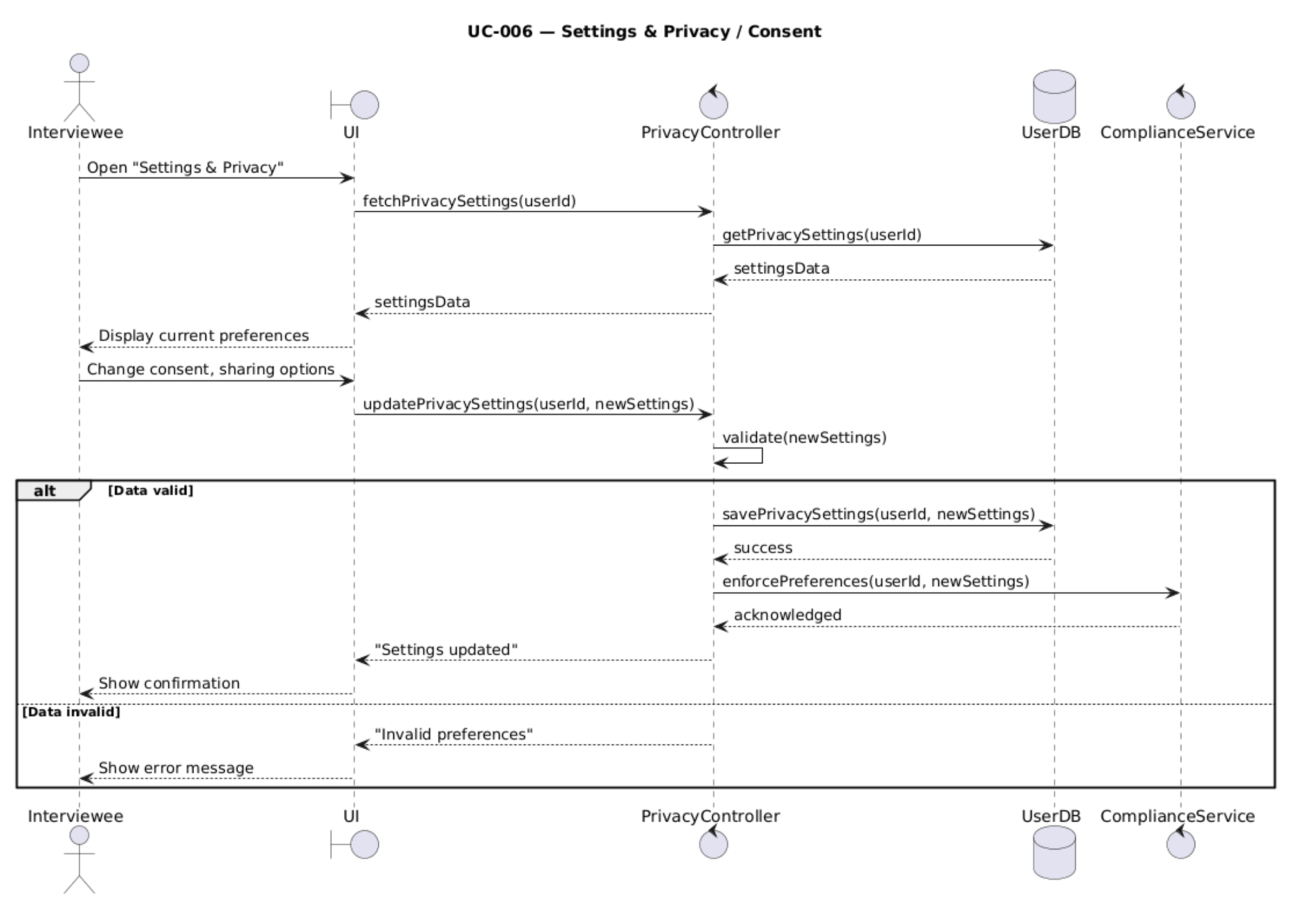
## Forgot/Reset Password



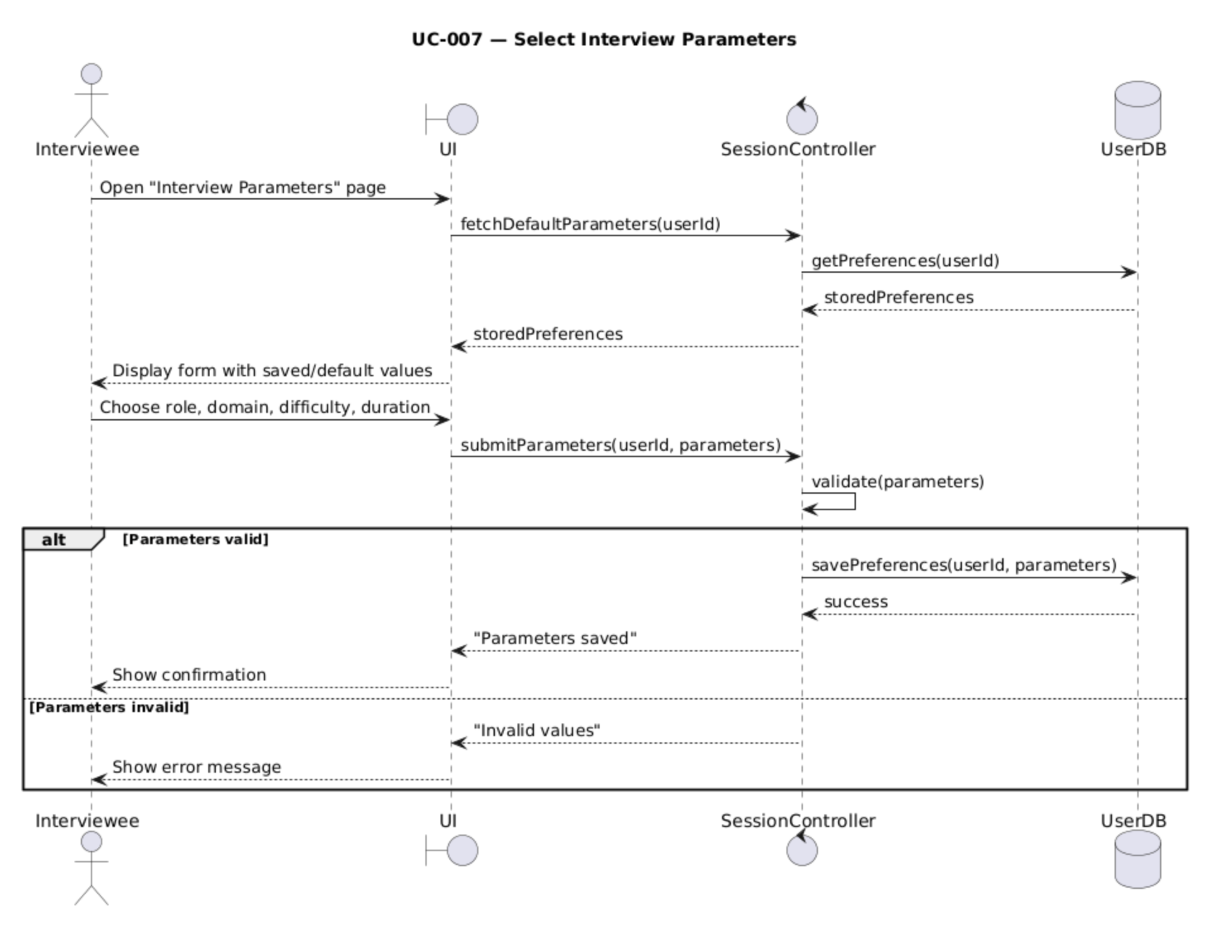
## Edit Profile



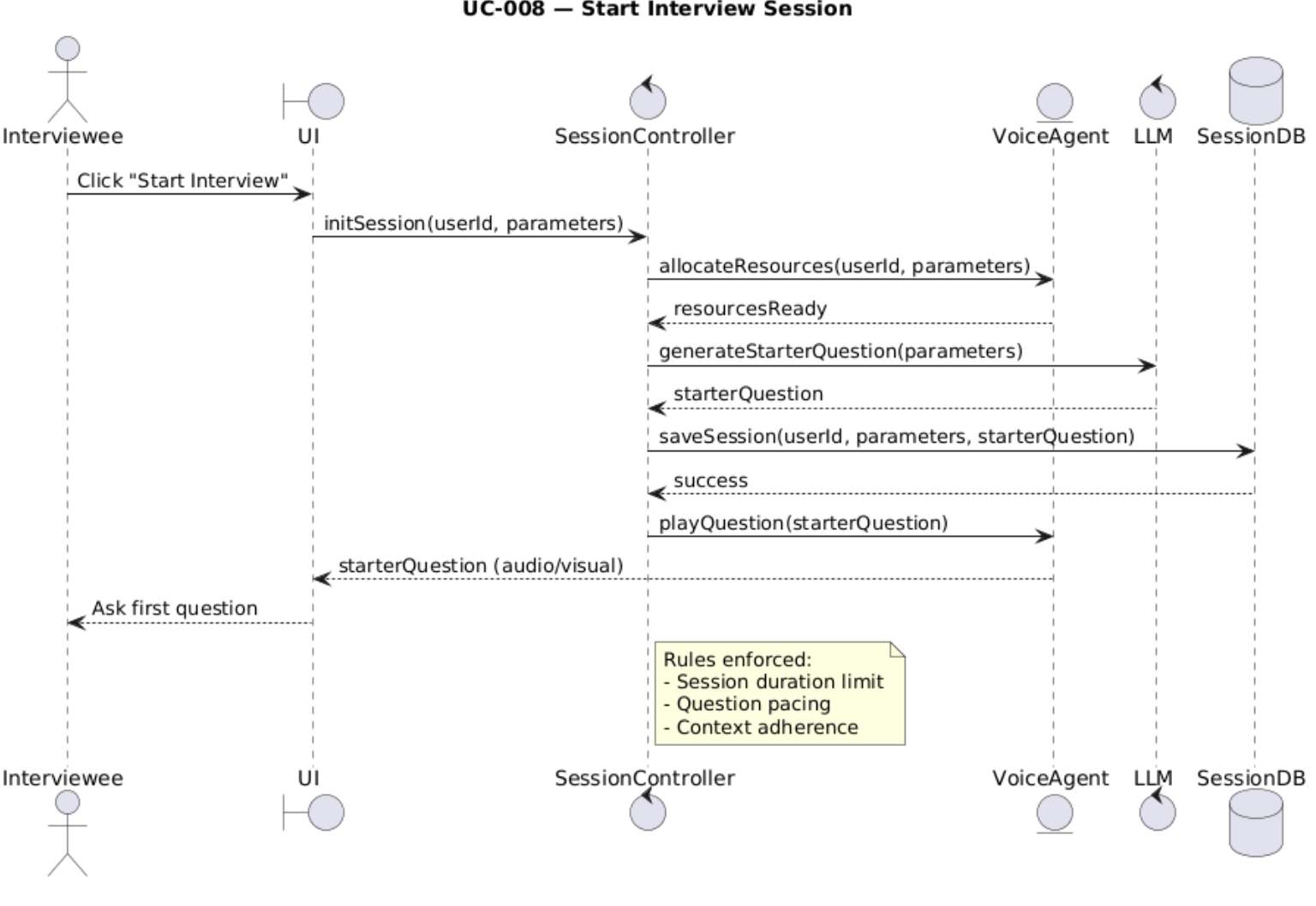
## Settings and Privacy



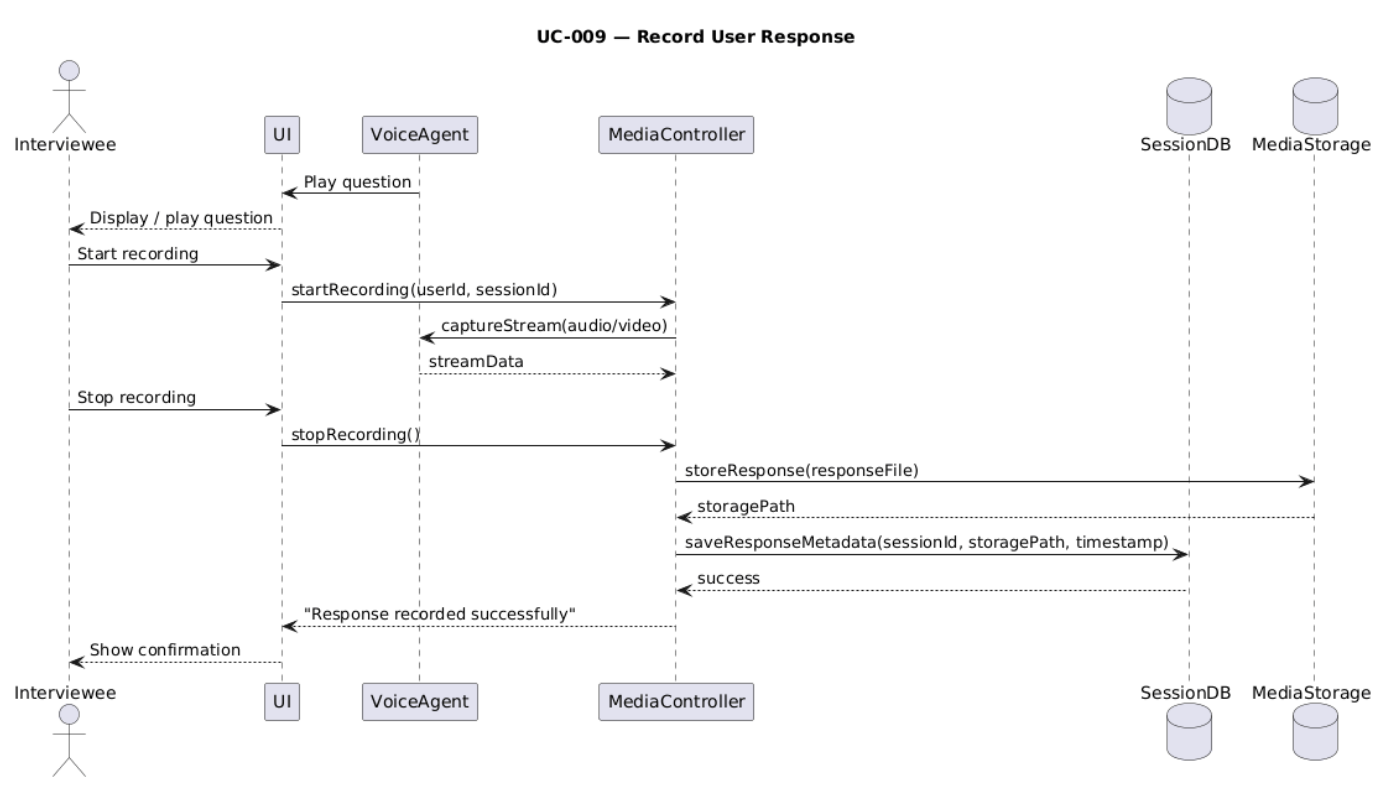
## Select interview parameters



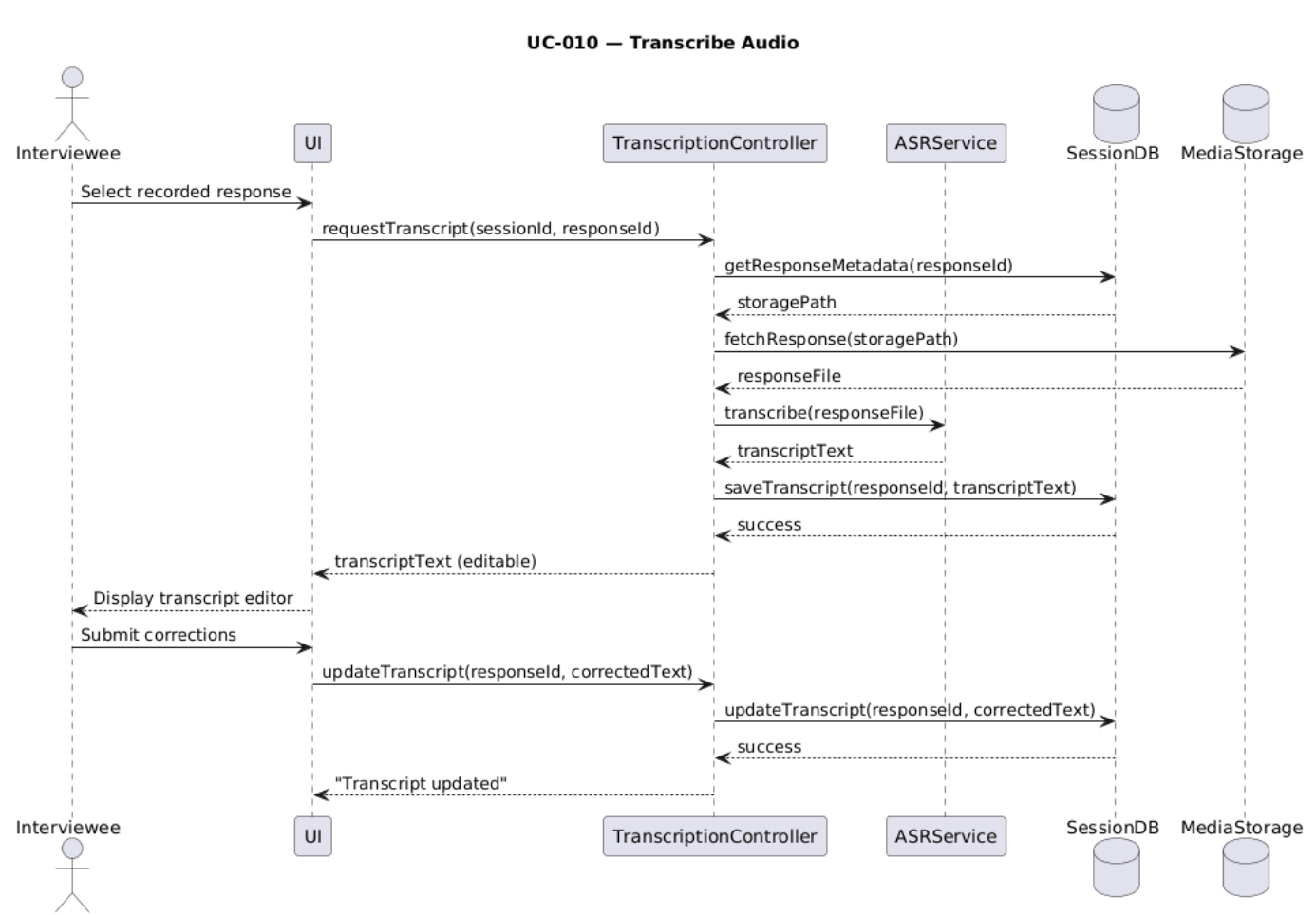
## Start interview Session



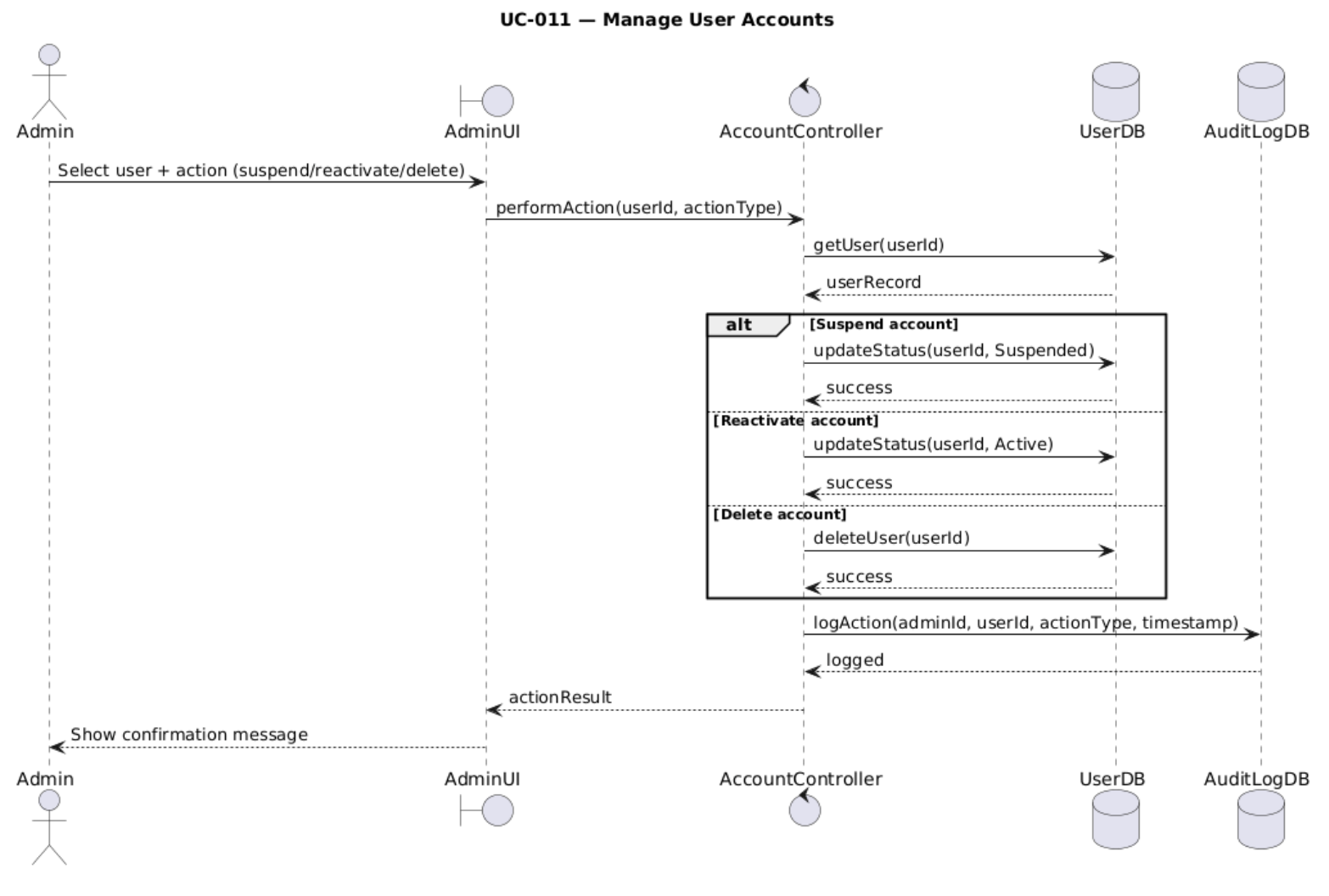
## Record User Response



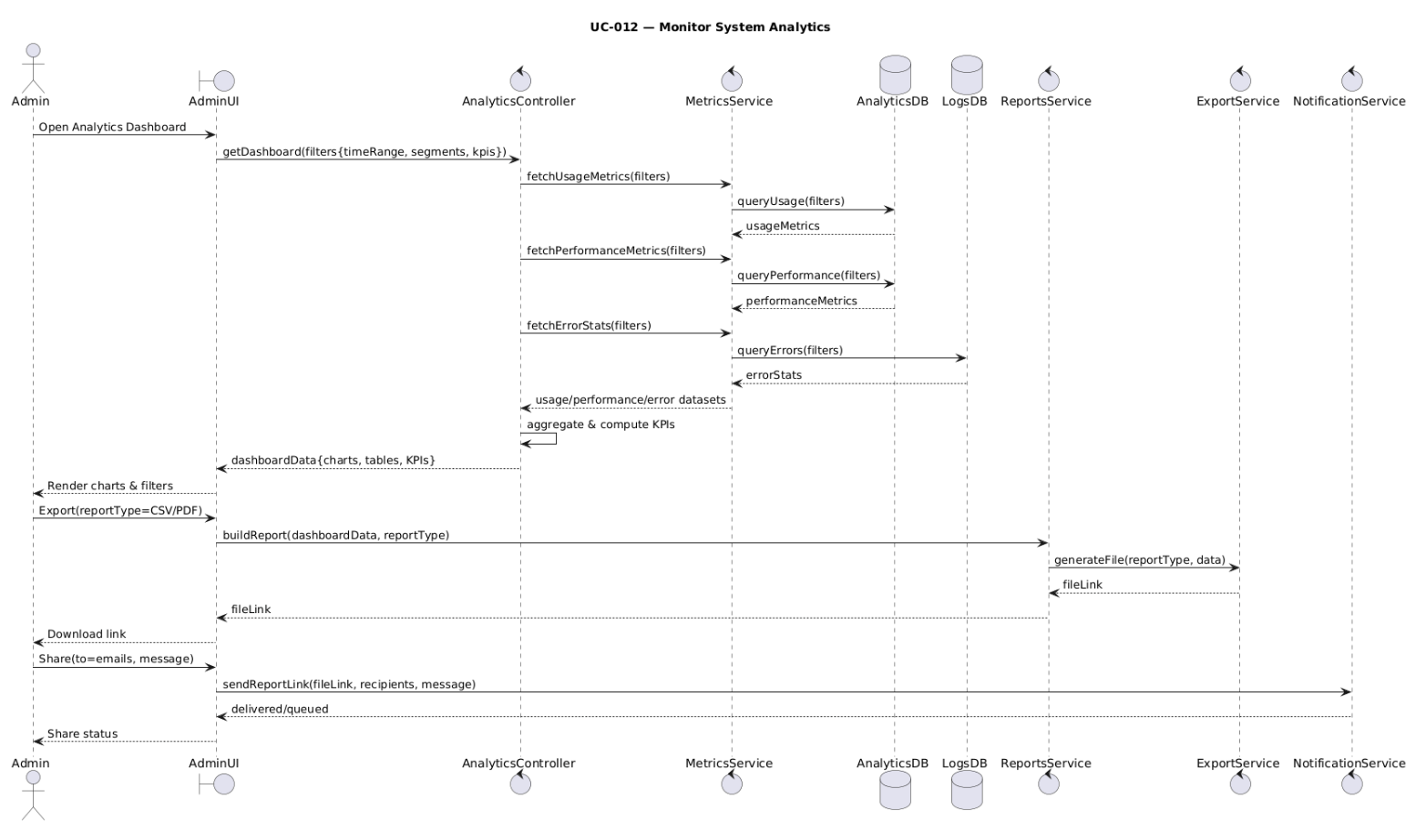
## Transcribe Audio



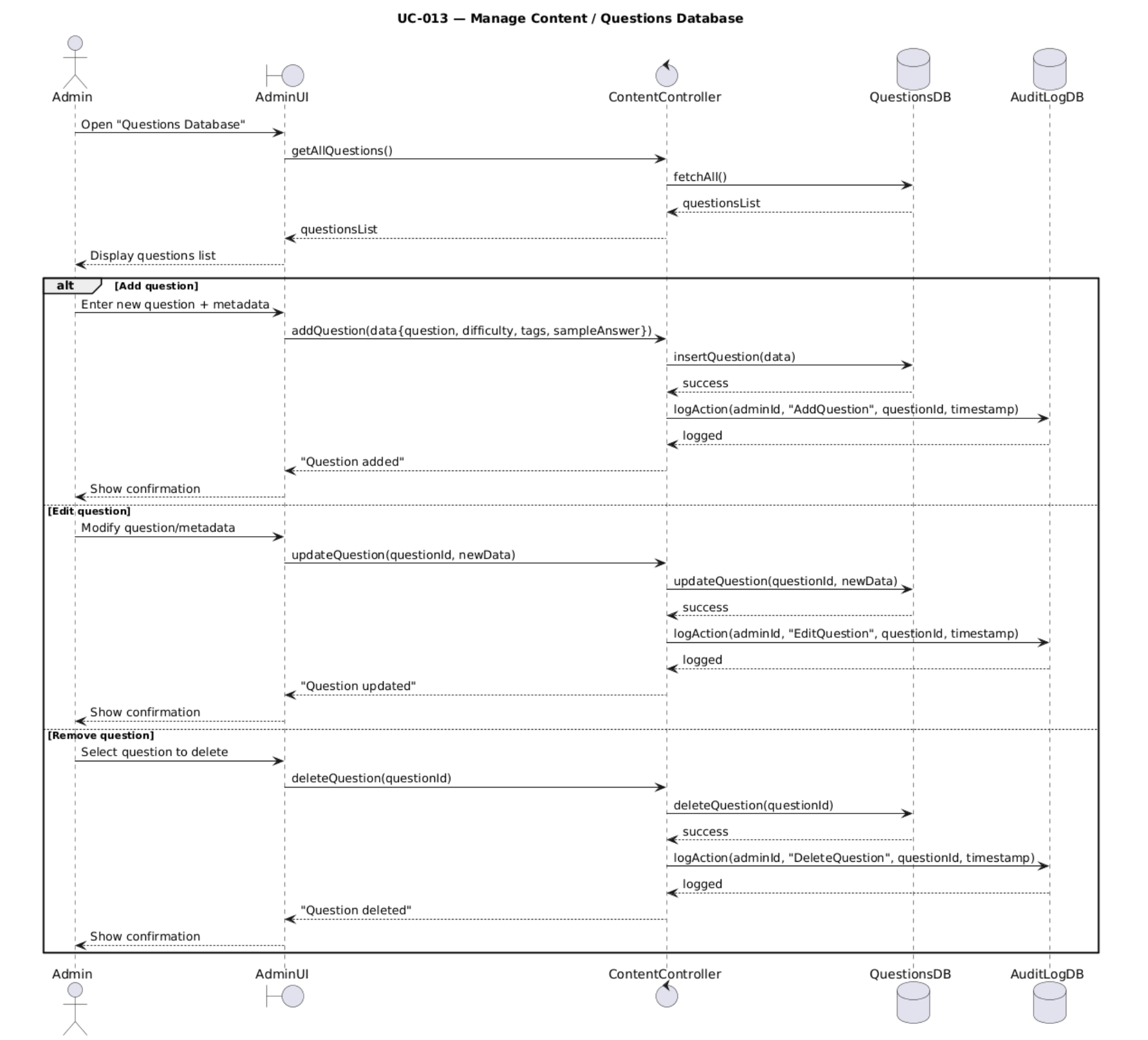
## Manage User accounts



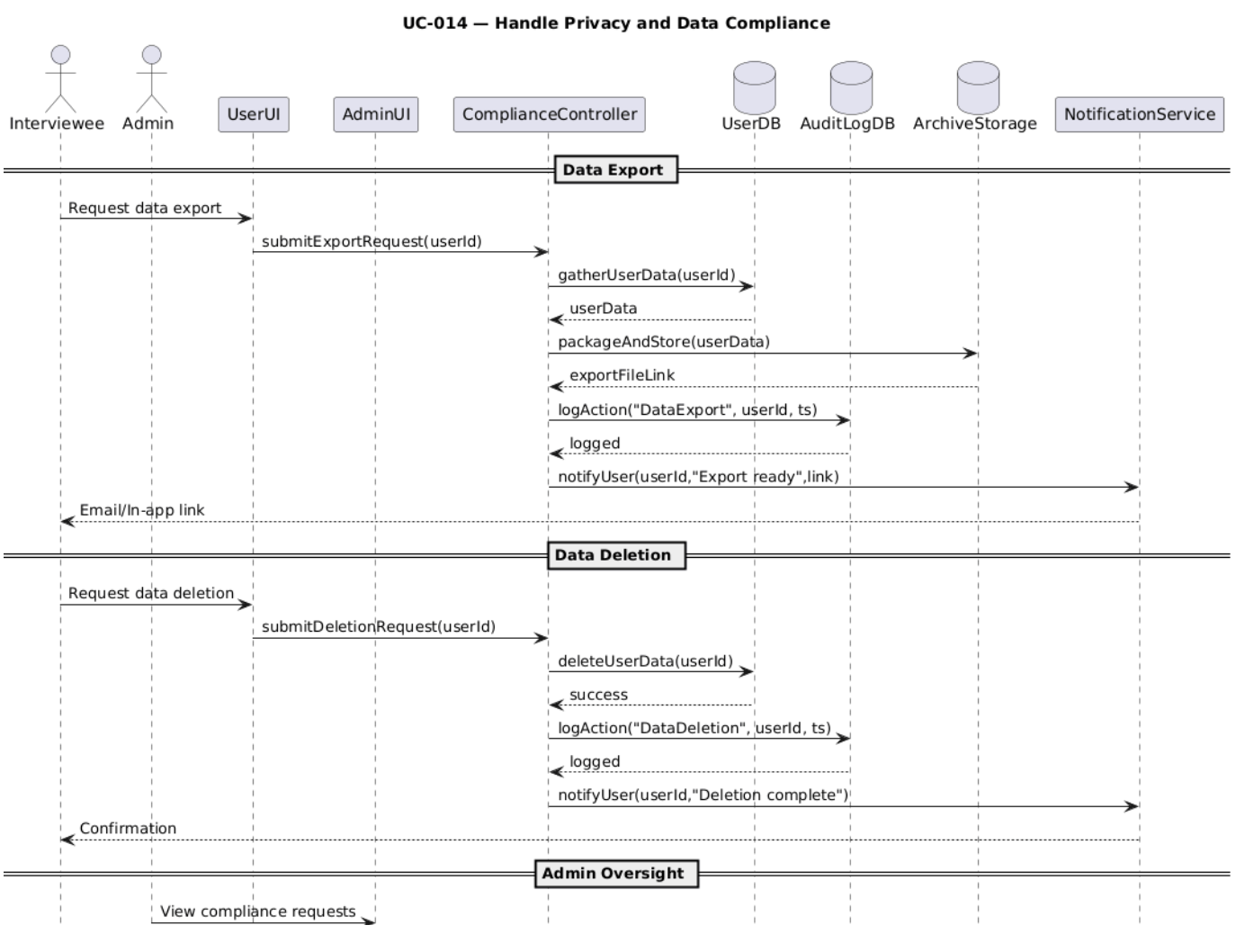
## Monitor System Analytics



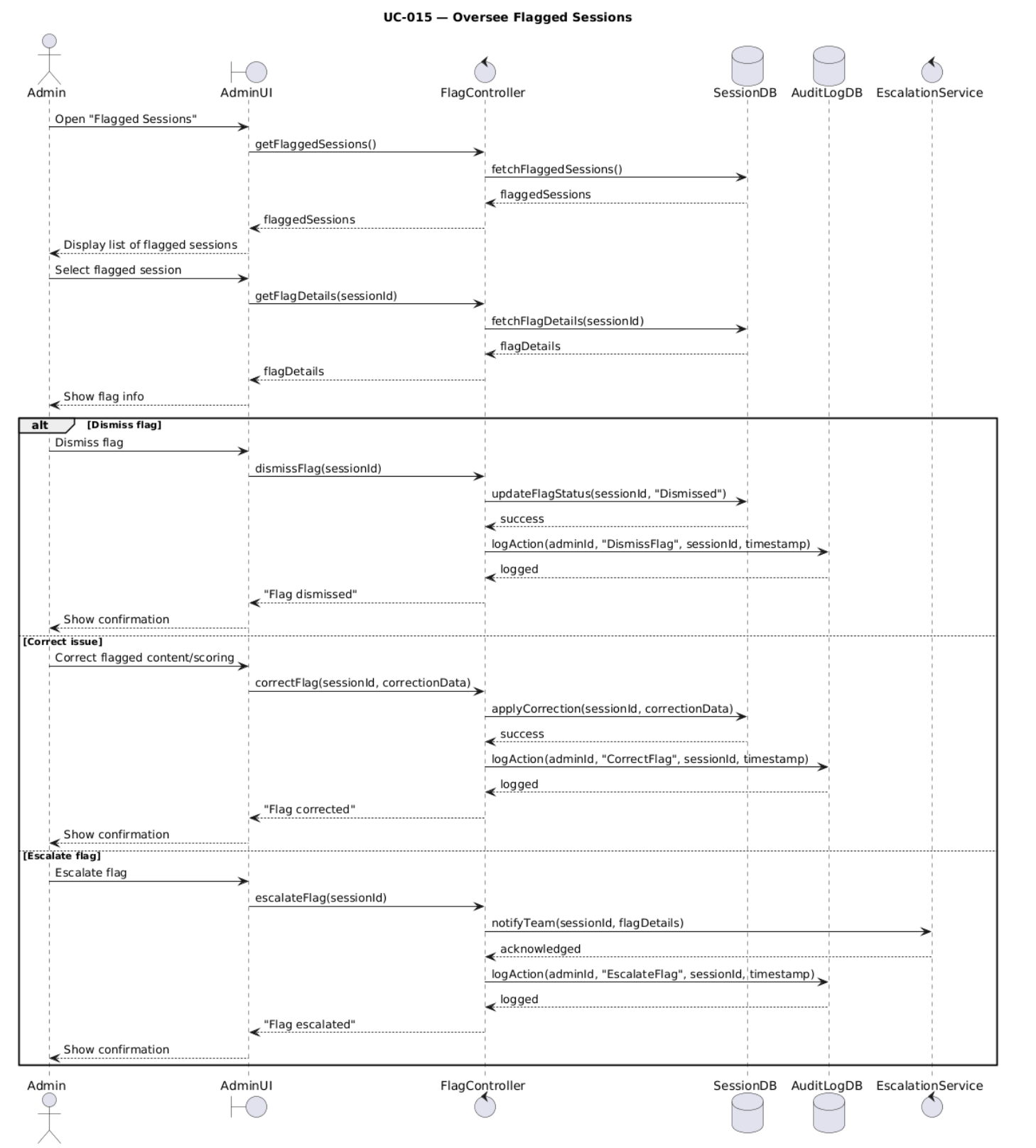
## Manage Content/Questions Database



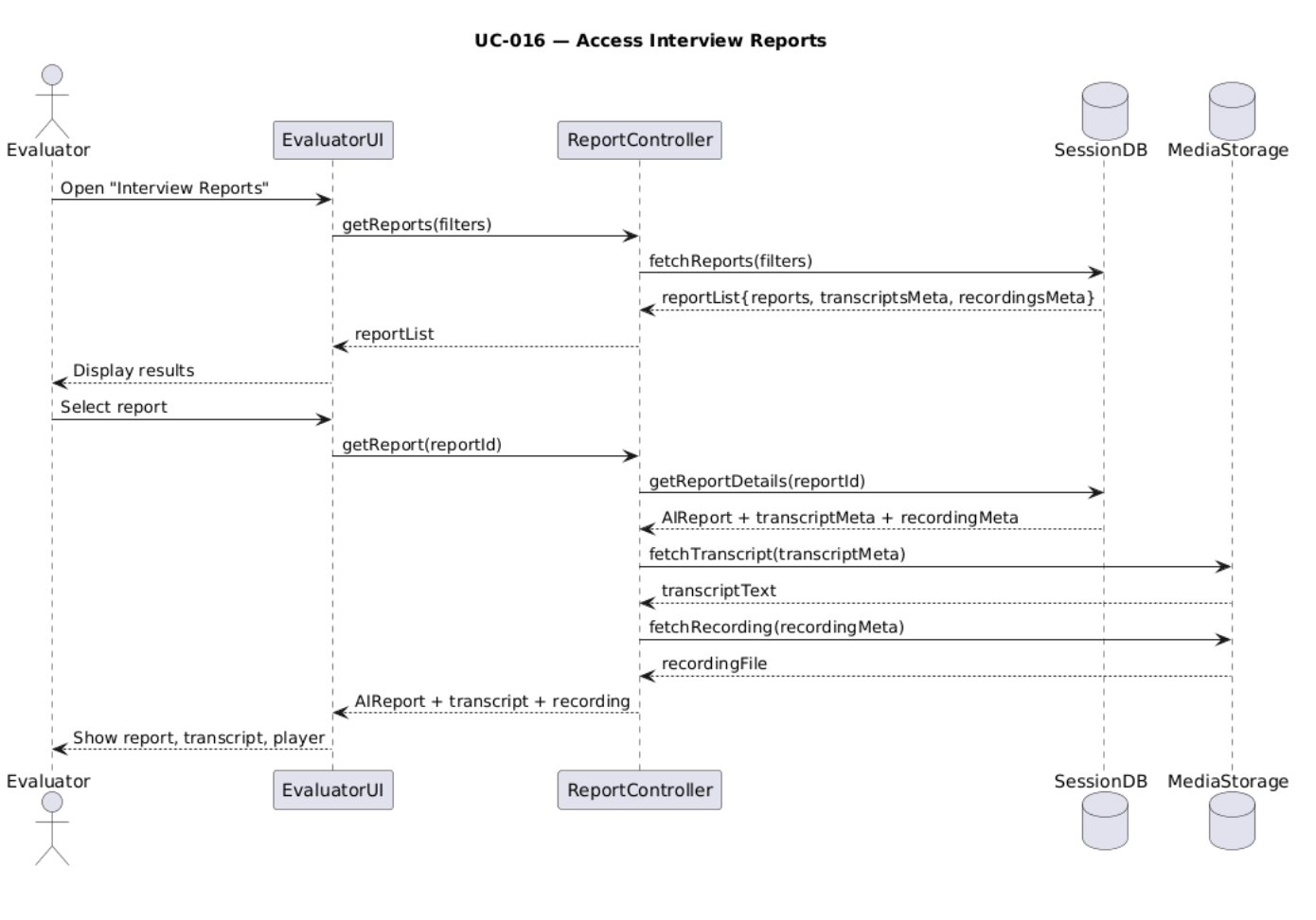
## Handle Privacy and Data compliance



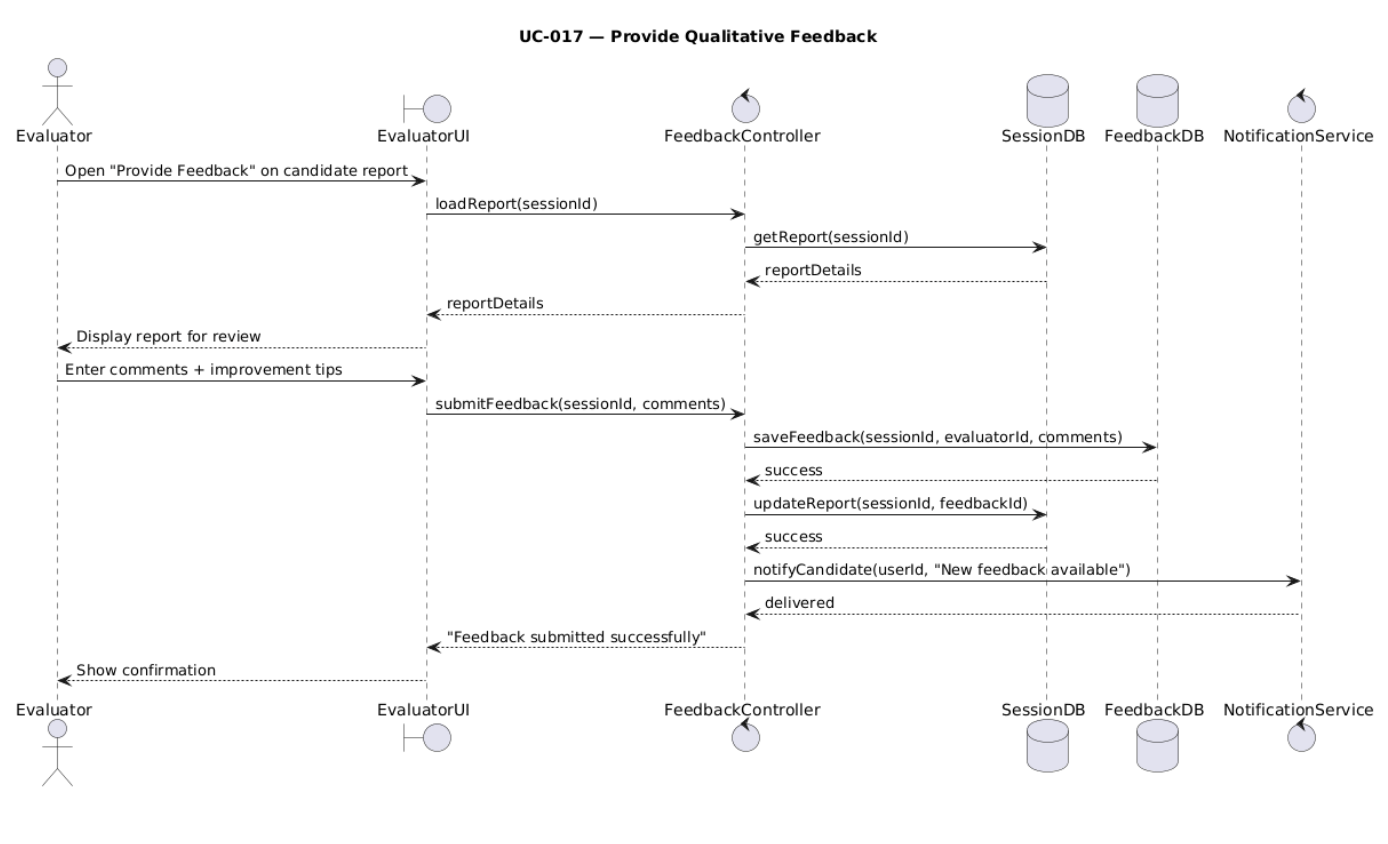
## Oversee Flagged Sessions



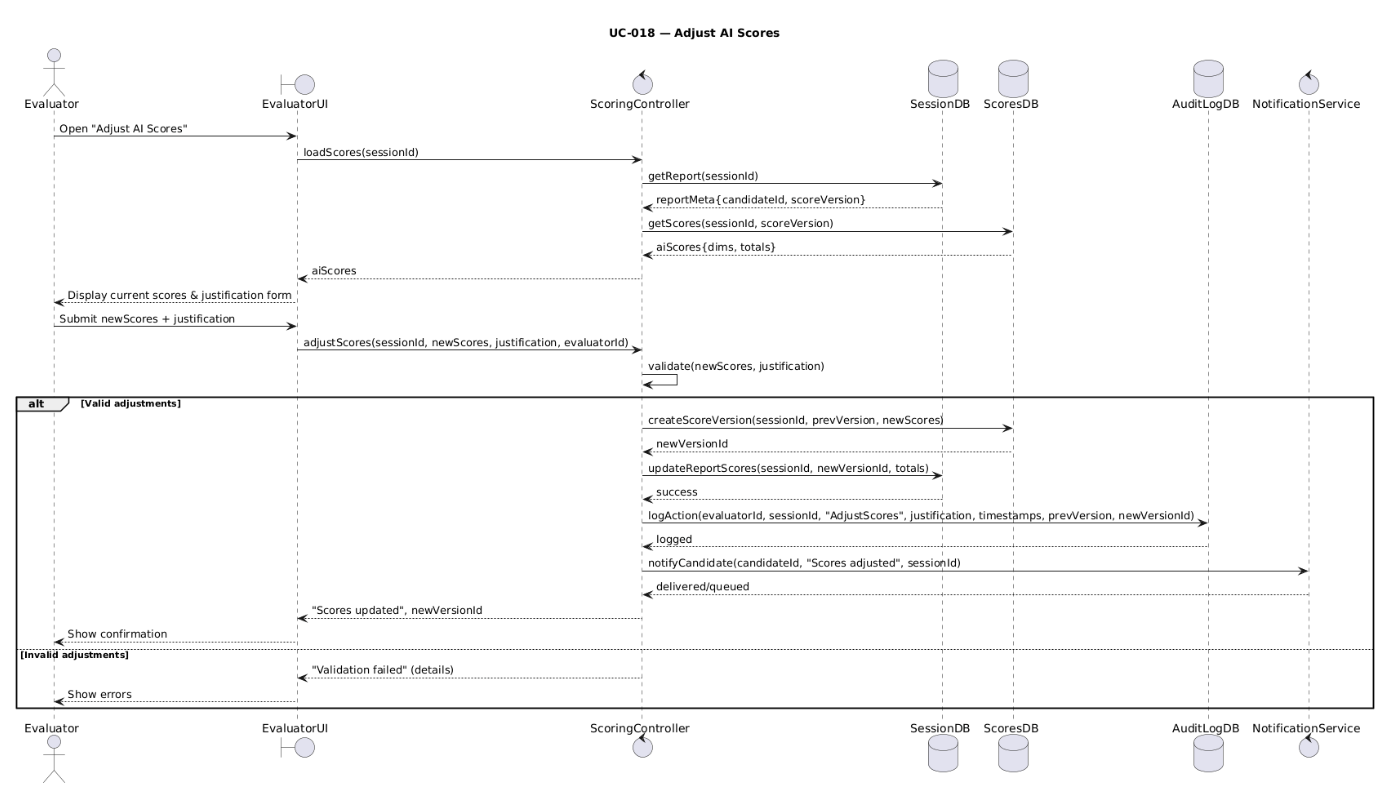
## Access Interview Reports



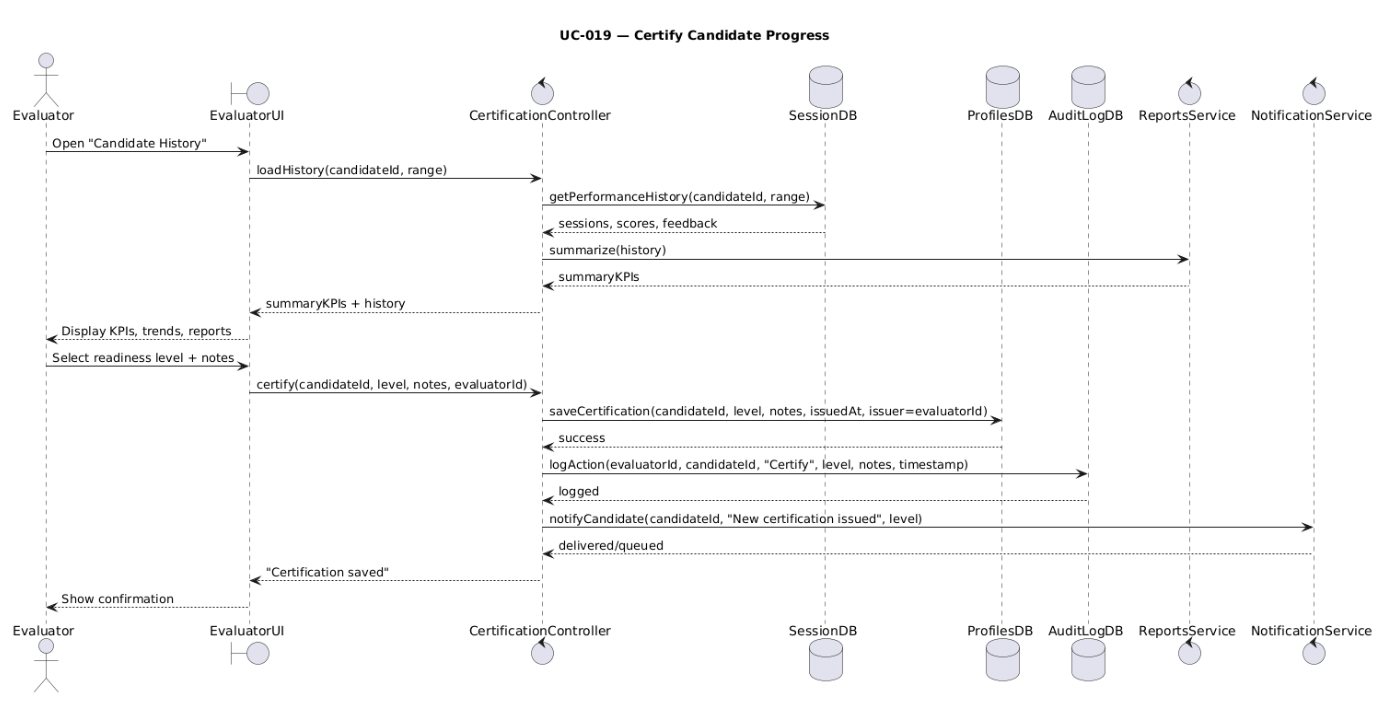
## Provide Qualitative Feedback



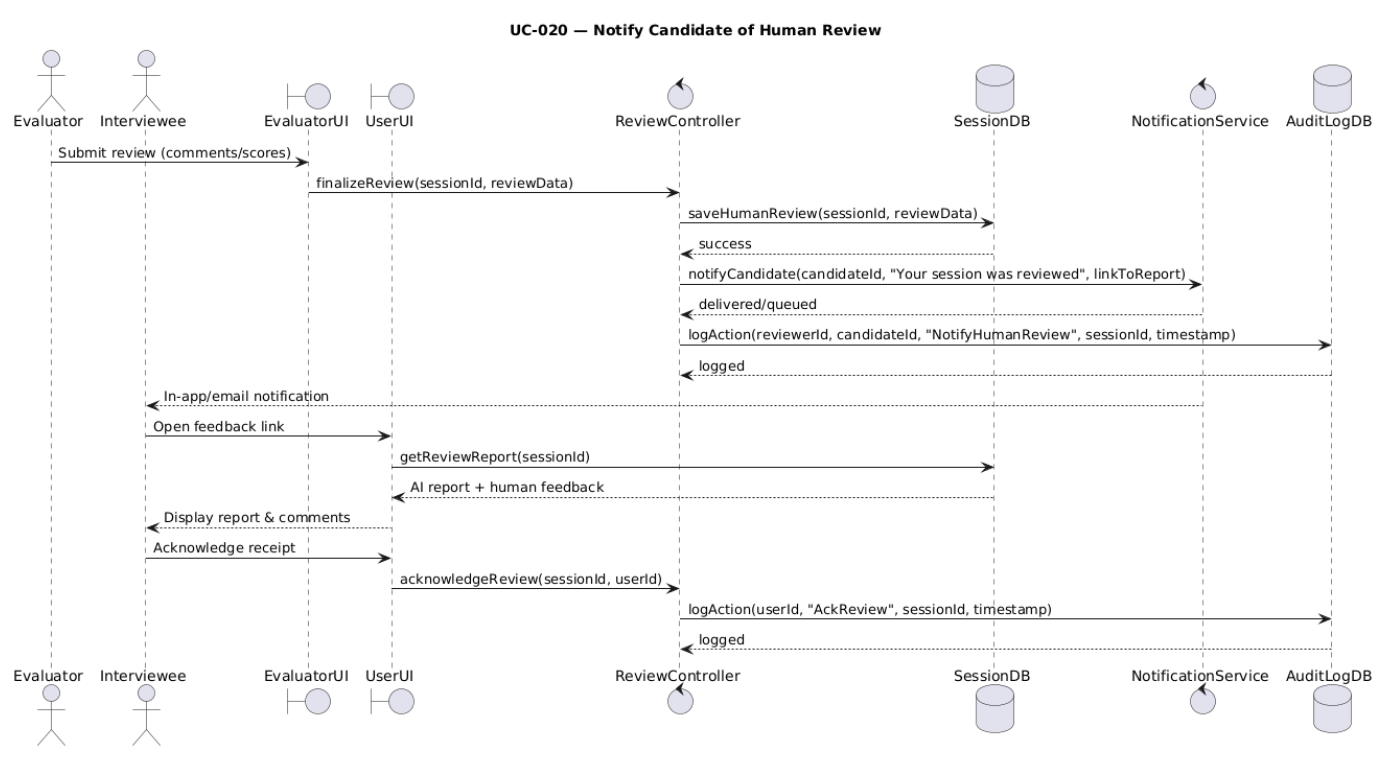
## Adjust Ai Scores



## Certify Candidate Progress



## Notify Candidate of Human Review



# State Diagrams

## Diagram details

**Object:** Interview Session

**Purpose:** Lifecycle of a single mock interview from context intake to feedback delivery.

**Primary states:**

* **DraftContext** (collecting job role/company; technical vs. behavioral focus)
* **Ready** (context validated; starter question generated & stored)
* **AskingStarter** (voice agent asks stored starter)
* **Listening** (candidate speaking via voice)
* **Transcribing** (speech → text via ASR)
* **GeneratingFollowUp** (LLM creates next question)
* **AskingFollowUp** (voice agent delivers follow-up; difficulty may adapt)
* **TimingOut** (session time limit reached)
* **Ended** (session closed; feedback generation triggered)
* **Error** (ASR/LLM or system failure; logged for audit)

**Events / messages:**

initContext, start, starter.ready, speech.start/stop, asr.ok/asr.fail, llm.ok/llm.fail, timeout, end, cancel. (Derived from user/system requirements on ASR/LLM integration, session timing, logging.)

**Notes/constraints:**

* First question within **5s** of job details; follow-ups within **2s**; report within **15s** after completion.

**Object:** Voice Agent

**Purpose:** Orchestrates the conversational flow (TTS/ASR turns), handles barge-in, and adapts difficulty based on performance.

**Primary states:**

* **Idle** (awaiting turn)
* **Prompting/Speaking** (asking starter or follow-up)
* **Capturing/Listening** (record user answer via voice)
* **Transcribing** (send audio to ASR)
* **AdaptingDifficulty** (policy adjusts next question difficulty)
* **Recovering** (retry after ASR/LLM failure; log event)
* **Error** (unrecoverable failure)

**Events / messages:**

beginTurn, tts.done, bargeIn, speech.stop, asr.ok/asr.fail, policy.decide(nextQ), recover, endTurn. (Aligned with dynamic generation and ASR integration.)

**Notes/constraints:**

* Must meet follow-up latency targets (**≤2s**).

**Object:** User Account

**Purpose:** Registration, authentication (incl. MFA), recovery, and lock/unlock flows.

**Primary states:**

* **Unregistered** (no account)
* **Registered–Unverified** (post-signup, pre-verification)
* **Verified** (profile manageable; can start sessions)
* **Authenticated** (active session post login)
* **Locked** (security lockout / admin action)

**Events / messages:**

register, verify.ok, login.ok/login.fail, mfa.ok, logout, reset.request, reset.complete, lock, unlock. (Auth + recovery + admin controls.)

**Notes/constraints:**

* **Broken Authentication** risk mitigated via MFA/JWT.

**Object:** Feedback Pipeline

**Purpose:** Aggregate session artifacts and produce a structured feedback report.

**Primary states:**

* **Pending** (await session end)
* **Aggregating** (ingest Q&A, timestamps, transcripts)
* **Scoring** (analyze clarity, confidence, correctness, STAR)
* **GeneratingReport** (compile structured report)
* **Stored** (persist report & session data securely)
* **Delivered** (available for revisit by candidate)
* **Error** (ingest/score/generation/storage failure; log)

**Events / messages:**

session.end, ingest.ok/fail, score.ok/fail, report.ok/fail, store.ok/fail, deliver.ok/fail.

**Notes/constraints:**

* The report must be ready within **15s** after interview completion.

**Object:** Admin Console

**Purpose:** Account moderation, parameter configuration, monitoring, and audits.

**Primary states:**

* **Idle**
* **ManagingAccounts** (create/suspend/delete candidates)
* **ConfiguringPolicy** (session duration, question complexity)
* **Monitoring** (performance/usage statistics; uptime/concurrency)
* **Auditing** (review logs & audit trails)
* **Error** (admin action failed)

**Events / messages:**

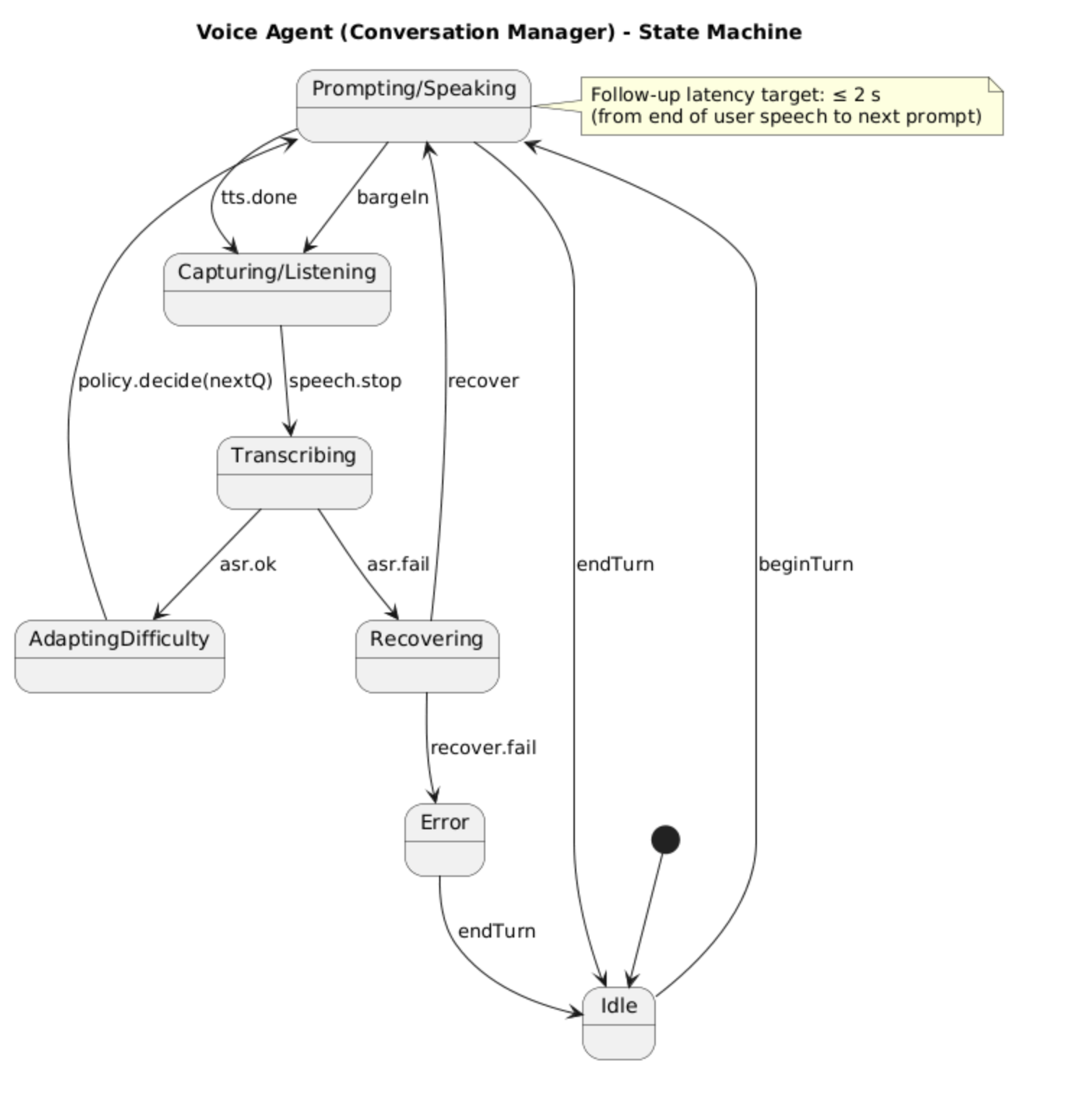
account.create/suspend/delete, policy.update, metrics.refresh, audit.view, audit.export, action.fail.

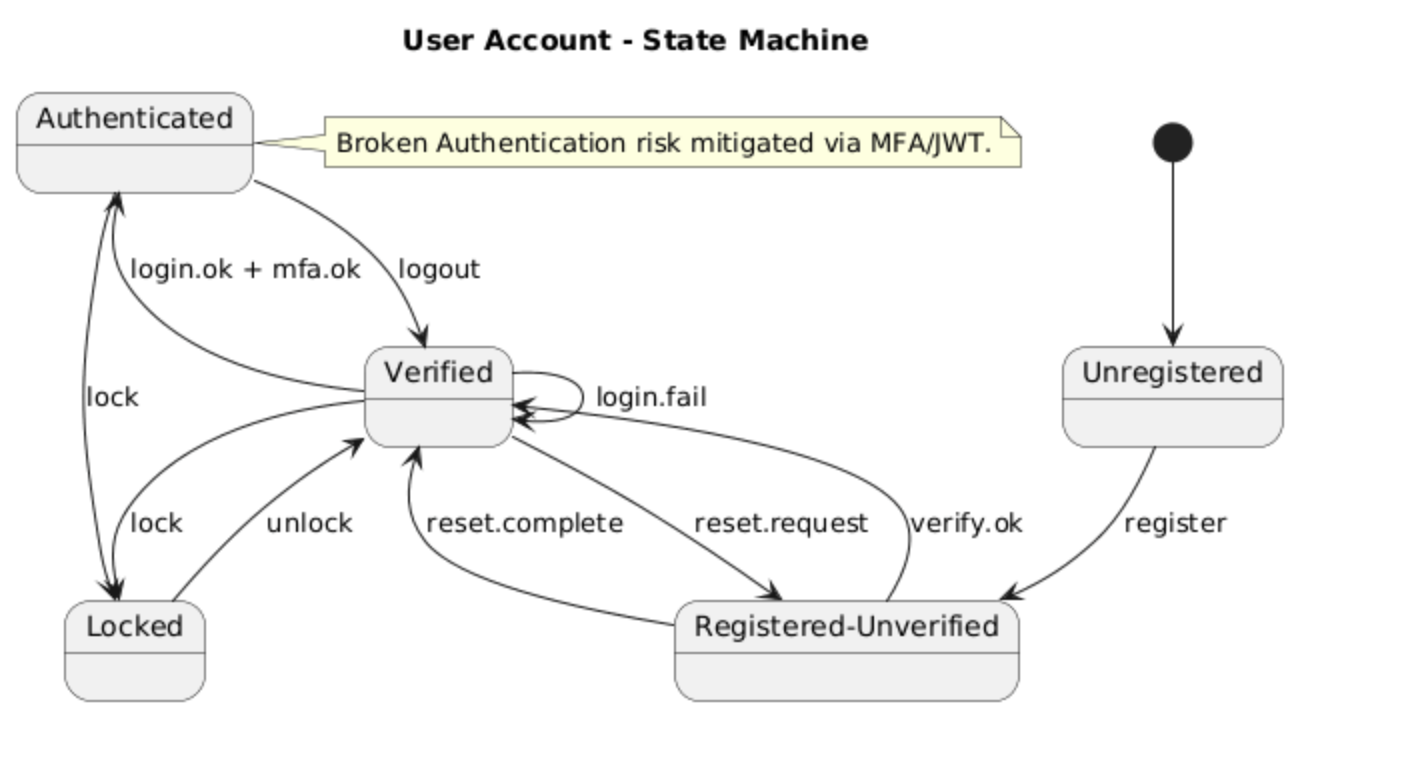
**Notes/constraints:**

* Security risks & controls apply (input validation, data provenance, supply-chain vetting, API validation).

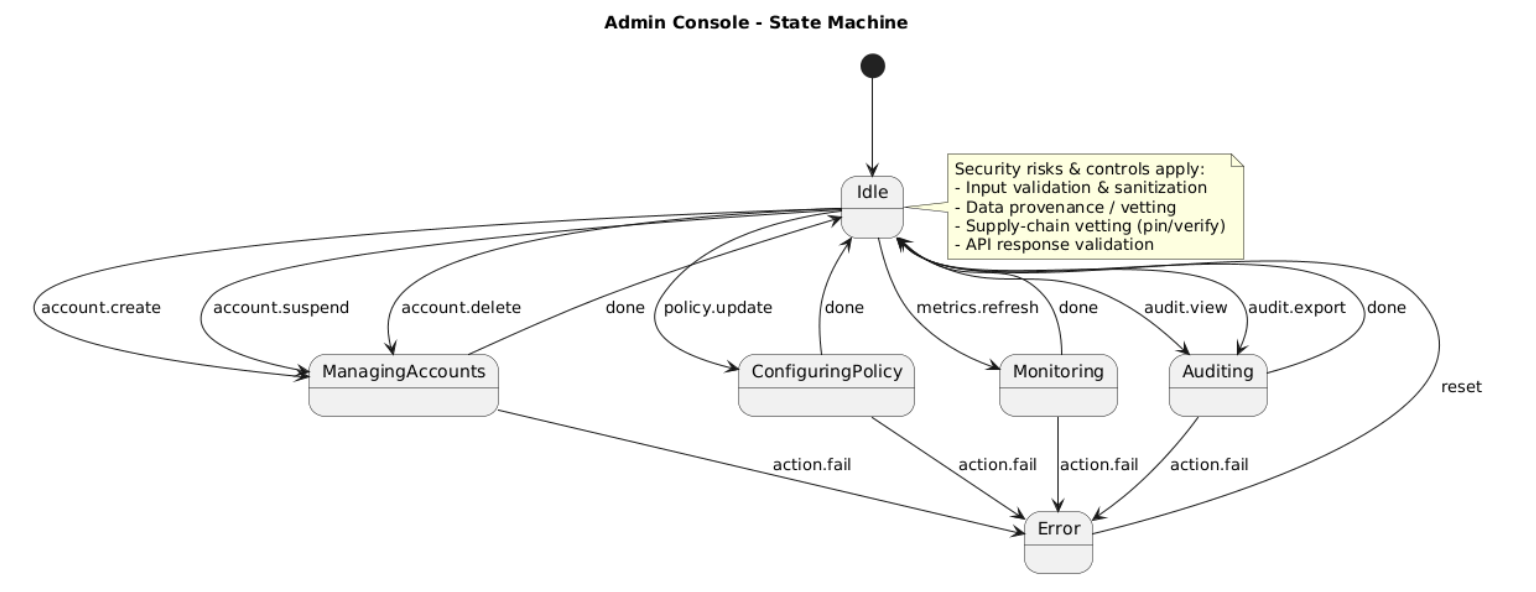
## Diagram











# Data Requirements

< If applicable, define the data needed for training, validation, and testing.>

| **Data Sources** | •Books, such as Cracking the Coding Interview)  • Online platforms: LeetCode, Codeforces  • Public datasets: CodeSearchNet (coding Q&A)  • Company-specific interview questions (reliable repositories) |
| --- | --- |
| **Data Requirements** | • Size: ~1–2k questions/answers (coding + behavioral + design)  • Missing values allowed: ≤5%  • Demographic attributes (e.g., gender, race) excluded as inputs  • Preprocessing: clean text, remove duplicates, anonymize sensitive info |
| **Model Requirements** | • Techniques: Transformer models (NLP), fine-tuned QA models  • Training: supervised fine-tuning, retrain monthly/after 1k new samples  • Accuracy target: ≥75% F1  • Validation: stratified split, early stopping |

# Non-functional Requirements / Quality Attributes

| **Sr#** | **Requirements** |
| --- | --- |
| 1 | The system shall generate the first interview question within 5 seconds after job details are provided. |
| 2 | The system shall generate follow-up questions within 2 seconds after a candidate’s response. |
| 3 | The system shall maintain 99.5% uptime and recover from crashes within 5 minutes. |
| 4 | All user data and interview sessions shall be stored with encryption at rest and in transit. |
| 5 | The system shall support up to 500 concurrent interview sessions without performance degradation. |
| 6 | The user interface shall be responsive and accessible on desktop and mobile devices. |
| 7 | The system shall provide a feedback report within 15 seconds after interview completion. |

# Security Requirements

| **Sr#** | **Security Risks** | **Potential Losses** | **Controls** |
| --- | --- | --- | --- |
| 1 | Input Manipulation Attack | Misleading feedback leading to loss of customers | Input validation & sanitization for all uploads |
| 2 | Data Poisoning Attack | Cost to retrain models and clean datasets | Strict data provenance and vetting |
| 3 | AI Supply Chain Attacks | Compromise of your app or models leading to data leaks | Use vetted sources, pin versions, verify checksums, and signatures for model binaries and packages. |
| 4 | Broken Authentication | Unauthorized access to personal data | Use robust authentication like MFA, JWT |
| 5 | Unrestricted Resource Consumption | Degraded performance for real users | Monitor resource usage and apply circuit breakers |
| 6 | Unsafe Consumption of APIs | Collateral breaches via third-party integrations | Validate all data from external APIs |

# 

# Security Engineer

| **Name of the Security Engineer** | Abdul Samad |
| --- | --- |

# Use of Generative AI

After writing the use cases and their descriptions, we used Gen AI to write uml code from it to make the diagrams

# Who Did What?

| **Name of the Team Member** | **Tasks done** |
| --- | --- |
| Abdul Rehman | 10 use cases, System Actors |
| Ahmad Bilal | 10 use cases, Sequence Diagram |
| Abdul Samad | Security requirements, Data Requirements |
| Haider Abbas Virk | Use Case Diagram, Class Diagram, Non-Functional Requirements |
| Muhammad Kabli | Introduction, State Diagram |

# Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

| **Section** **Title** | **Reviewer Name(s)** |
| --- | --- |
| Introduction, System Actors, | Abdul Rehman |
| Use Cases | Ahmad Bilal |
| Class Diagram, Sequence Diagram, | Abdul Samad |
| State Diagram, Data Requirements | Muhammad Kabli |
| Non Functional Requirement, Security Requirements | Haider Abbas Virk |