|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| **Capstone Project Document** |
| **Cabin Crew Recruitment and Promotion Management System for Aviation Academy** |

|  |  |
| --- | --- |
| **GFA25SE113** | |
| **Group Members** | Trương Thành Nhân - SE151229  Trần Duy Hưng - SE173089  Huỳnh Nguyễn Ngọc Nguyên - SE171880  Trần Nhật Hoàng - SE171204  Hoàng Lê Nhật Minh - SE171857 |
| **Supervisor** | **Ngô Đăng Hà An** |
| **Ext Supervisor** |  |
| **Capstone Project code** | **FA25SE204** |

- Ho Chi Minh, December 2025 -

**Table of Contents**

[Acknowledgements 5](#_Toc3771)

[Definition and Acronyms 6](#_Toc27401)

[I. Project Introduction 7](#_Toc24731)

[1. Overview 7](#_Toc10535)

[1.1 Project Information 7](#_Toc14768)

[1.2 Project Team 7](#_Toc22459)

[2. Product Background 7](#_Toc9238)

[3. Existing Systems 8](#_Toc19266)

[3.1 Vietnam Airlines e-Recruitment Portal 8](#_Toc29179)

[3.2 Bamboo Airways Career Website 8](#_Toc9798)

[4. Business Opportunity 9](#_Toc22640)

[5. Software Product Vision 10](#_Toc15235)

[6. Project Scope & Limitations 10](#_Toc12675)

[6.1 Major Features 10](#_Toc3591)

[6.2 Limitations & Exclusions 11](#_Toc10336)

[II. Project Management Plan 12](#_Toc18265)

[1. Overview 12](#_Toc26448)

[1.1 Scope & Estimation 12](#_Toc31737)

[1.2 Project Objectives 14](#_Toc24263)

[1.3 Project Risks 15](#_Toc6319)

[2. Management Approach 16](#_Toc13805)

[2.1 Project Process 16](#_Toc26501)

[2.2 Product Backlog 17](#_Toc14390)

[2.3 Quality Management 17](#_Toc20835)

[2.4 Training Plan 18](#_Toc28944)

[3. Project Deliverables 18](#_Toc27036)

[4. Responsibility Assignments 18](#_Toc28080)

[5. Project Communications 19](#_Toc31482)

[6. Configuration Management 19](#_Toc7804)

[6.1 Document Management 19](#_Toc18948)

[6.2 Source Code Management 19](#_Toc23729)

[6.3 Tools & Infrastructures 19](#_Toc7826)

[III. Software Requirement Specification 20](#_Toc32295)

[1. Product Overview 20](#_Toc30447)

[2. User Requirements 21](#_Toc28782)

[2.1 Actors 21](#_Toc6423)

[2.2 Use cases 22](#_Toc23960)

[3. Functional Requirements 26](#_Toc6866)

[3.1 System Functional Overview 26](#_Toc12383)

[3.2 Functional Requirements 38](#_Toc32698)

[4. Non-Functional Requirements 44](#_Toc14996)

[4.1 External Interfaces 44](#_Toc17524)

[4.2 Quality Attributes 45](#_Toc6139)

[5. Requirement Appendix 45](#_Toc10399)

[5.1 Business Rules 45](#_Toc30969)

[5.2 Common Requirements 50](#_Toc21830)

[5.3 Application Messages List 51](#_Toc7107)

[5.4 Other Requirements… 51](#_Toc11222)

[IV. Software Design Description 52](#_Toc12266)

[1. System Design 52](#_Toc10738)

[1.1 System Architecture 52](#_Toc32641)

[1.2 Package Diagram 52](#_Toc28813)

[1.3 Activity Diagrams 54](#_Toc21036)

[1.4 State Machine Diagrams 74](#_Toc12598)

[2. Detailed Design 74](#_Toc25042)

[2.1 Create Campaign Request 74](#_Toc6788)

[2.2 Create Campaign 75](#_Toc13776)

[2.3 Campaign Task’s Assignment 76](#_Toc25843)

[2.4 Apply To Campaign Round 77](#_Toc8418)

[2.5 Create Appearance Evaluation Form 78](#_Toc27585)

[2.6 Create Interview Evaluation Form 79](#_Toc16147)

[2.7 Take English Test 80](#_Toc22913)

[3. Database Design 81](#_Toc31863)

[3.1 Diagram 81](#_Toc28292)

[3.2 Tables Description 81](#_Toc12930)

[V. Software Testing Documentation 99](#_Toc22342)

[1. Scope of Testing 99](#_Toc6449)

[2. Test Strategy 99](#_Toc31613)

[2.1 Testing Types 99](#_Toc29282)

[2.2 Test Levels 99](#_Toc12935)

[2.3 Supporting Tools 99](#_Toc18704)

[3. Test Plan 99](#_Toc28733)

[3.1 Human Resources 99](#_Toc11086)

[3.2 Test Environment 99](#_Toc21885)

[3.3 Test Milestones 99](#_Toc15670)

[4. Test Cases 99](#_Toc10731)

[5. Test Reports 99](#_Toc13346)

[VI. Release Package & User Guides 99](#_Toc12460)

[1. Deliverable Package 99](#_Toc1262)

[2. Installation Guides 99](#_Toc6200)

[2.1 System Requirements 99](#_Toc9496)

[2.2 Installation Instruction 99](#_Toc1392)

[3. User Manual 99](#_Toc32238)

[3.1 Overview 99](#_Toc31892)

[3.2 Workflow 1 99](#_Toc27185)

[3.3 Workflow 2 99](#_Toc16089)

# Acknowledgements

We would like to express our deepest gratitude and thanks to all individuals and organizations who don't hesitate to take the time to support, advise and guide us throughout the process of making this project.

Foremost, we would like to express our sincere gratitude to our mentor Mr. Ngo Dang Ha An for the continuous support of our capstone project, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped us in all the time of research, designing, building, and developing of this thesis. We could not have imagined having a better advisor and mentor for our capstone project.

Besides our advisor, we would like to thank the rest of our thesis committee: Mr. Dang Ngoc Minh Duc, Mr. Nguyen Tan Phuc Mr. , for their quality recommend, insightful comments, and challenging questions.

Last but not the least, we would like to thank our family for the overwhelming encouragements and supporting us spiritually throughout our life

# Definition and Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| PWM | Psychology website |
| AWS | Amazon Web Services |
| BA | Business Analysis |
| BR | Business Rule |
| ERD | Entity Relationship Diagram |
| GUI | Graphical User Interface |
| PM | Project Manager |
| SDD | Software Design Description |
| SPMP | Software Project Management Plan |
| SRS | Software Requirement Specification |
| UAT | User Acceptance Test |
| UC | Use Case |
| API | Application Program Interface |

# I. Project Introduction

## 1. Overview

### 1.1 Project Information

* Project name: CCRPMS – Cabin Crew Recruitment and Promotion Management System for Aviation Academy
* Project code: FA25SE204
* Group name: GFA25SE113
* Software type: Web application, Mobile Application

### 1.2 Project Team

#### 1.2.1 Supervisor

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Email** | **Phone Number** | **Title** |
| Ngô Đăng Hà An | anndh2@fpt.edu.vn | 0906782333 | Lecturer |

#### 1.2.2 Team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Email** | **Phone Number** | **Role** |
| Trương Thành Nhân | nhanttse151220@fpt.edu.vn | 0903348731 | Leader |
| Trần Duy Hưng | hungtdse173089@fpt.edu.vn | 0934753389 | Member |
| Huỳnh Nguyễn Ngọc Nguyên | nguyenhnnse171880@fpt.edu.vn | 0967634699 | Member |
| Trần Nhật Hoàng | hoangntse171204@fpt.edu.vn | 0343417544 | Member |
| Hoàng Lê Nhật Minh | minhhlnse171857@fpt.edu.vn | 0902855644 | Member |

## 2. Product Background

In the modern aviation industry, cabin crew members not only perform passenger service duties but also represent the airline’s brand image and directly influence the customer experience. Consequently, the recruitment and promotion processes for cabin crew must adhere to stringent standards, encompassing professionalism, effective communication, foreign language proficiency, and the ability to respond flexibly to situational challenges.

However, at the Aviation Academy, recruitment and promotion activities are still largely conducted manually, resulting in multiple limitations:

* **Traditional application submission:** Candidates submit paper-based applications or send unstandardized email forms.
* **Unsystematic interview process:** Candidate evaluation lacks standardization and is highly dependent on individual assessors.
* **Fragmented and dispersed record-keeping:** Candidate information is stored across multiple sources without centralized management.

These limitations generate several consequences:

* **Complex and time-consuming processes:** HR staff must manually enter data, verify applications, and respond individually to each candidate.
* **High risk of errors:** Applications may be misplaced, data duplicated, or evaluation results recorded inaccurately.
* **Lack of transparency for candidates:** Applicants cannot track their progress and must contact HR directly for updates.
* **Difficulties in data analysis and reporting:** Compiling results across multiple examinations and interviews requires significant time and human resources.
* **Lagging behind digital transformation trends:** While international airlines have adopted e-recruitment platforms, the Academy continues to rely on manual methods, reducing its competitiveness.

**Conclusion:** These issues highlight the urgent need for a comprehensive digital solution. The **Cabin Crew Recruitment and Promotion Management System (CCRPMS)** is therefore proposed to automate and standardize the process, improving management efficiency, minimizing errors, enhancing transparency, and supporting the Academy in advancing towards sustainable digital transformation.

## 3. Existing Systems

### 3.1 Vietnam Airlines e-Recruitment Portal

Description: Provides an online platform for application submission and candidate result notifications.

**Pros:**

* User-friendly interface that facilitates application submission.
* Supports status updates throughout the recruitment process.

**Cons:**

* Primarily focused on new recruitment; lacks a dedicated module for promotion management.
* Does not incorporate multidimensional evaluation tools (e.g., appearance, communication skills, professional competencies).

### 3.2 Bamboo Airways Career Website

Description: Allows candidates to apply online for cabin crew and pilot positions.

Pros:

* Simple, accessible design that connects applicants directly with the HR department.
* Widely used in aviation industry, well-established reputation across the globes

Cons:

* Does not support end-to-end management of the candidate life-cycle.
* Lacks comprehensive reporting and advanced data analytics capabilities.

**Observation:** Current systems address only specific aspects of recruitment but fail to deliver a holistic solution that integrates recruitment, promotion, and full life-cycle management for cabin crew. This gap underscores the strategic positioning of the Cabin Crew Recruitment and Promotion Management System (CCRPMS).

## 4. Business Opportunity

The aviation industry is currently undergoing rapid growth, accompanied by an increasing demand for both new cabin crew recruitment and promotion. Implementing the CCRPMS not only addresses existing procedural shortcomings but also generates strategic advantages for the Academy across the following dimensions:

* Process standardization and optimization
  + Consolidates all stages, including application intake, evaluation, interviews, promotion examinations, and reporting.
  + Reduces dependency on subjective human factors, ensuring transparent and consistent operations.
* Time and cost efficiency
  + Minimizes paperwork and manual tasks for HR departments.
  + Shortens application processing time and candidate response cycles.
  + Increases productivity without requiring additional resources.
* Enhanced candidate experience
  + Online portals allow candidates to track application status, receive interview notifications, access test schedules, and view transparent results.
  + Contributes to building a professional and modern institutional image.
* Transparency and reliable data
  + Systematic storage of all candidate results and evaluations.
  + Facilitates monitoring, auditing, and rapid data retrieval, thereby reinforcing trust and accountability.
* Scalability and external collaboration
  + The system can be expanded to serve multiple airlines in the future.
  + Positions the Academy as a strategic partner in aviation training and human resource management.
* Alignment with digital transformation trends
  + Enables the Academy to align with global practices by adopting advanced technologies in HR.
  + Enhances competitiveness and strengthens institutional positioning within the aviation sector.

**Conclusion:** The implementation of CCRPMS represents not only a technological upgrade but also a strategic initiative that enables the Academy to:

* Improve internal efficiency
* Enhance recruitment and promotion quality
* Unlock new collaboration and business opportunities
* And move closer to the goal of comprehensive digital transformation in aviation human resource management

## 5. Software Product Vision

**“CCRPMS aims to become a comprehensive cabin crew recruitment and promotion management system that enables the Aviation Academy to optimize processes, enhance transparency, improve candidate experience, and support data-driven decision-making.”**

Through the system, candidates can conveniently submit applications, track their progress, and participate in online assessments or interviews. Meanwhile, HR staff and evaluators are equipped with digital tools for assessment, scheduling, and statistical reporting.

## 6. Project Scope & Limitations

### **6.1 Major Features**

#### 6.1.1 Candidate uses Web Application

As a Candidate, I want to use these following features:

**FE-01.** Manage candidate account: create account, view profile, update profile

**FE-02.** Manage application: create application, view application, update application, delete application

**FE-03.** Manage recruitment process: view ongoing recruitment, view recruitment history

**FE-04.** Search ongoing campaign by category: airline partners, type of campaign

**FE-05.** Receive result from system, view result of a round

**FE-06.** View required English listening/speaking tests, take tests

**FE-07.** Manage enquiry request: create enquiry request, view status of enquiry request

**FE-08.** Apply application to an ongoing campaign

#### 6.1.2 Recruiter uses Web Application

As a Recruiter , I want to use these following features:

**FE-09.** Mange recruiter account: view profile, update profile

**FE-10.** Manage application: view application, approve application, reject application

**FE-11.** View assigned campaigns, view candidate/cabin crew list of a round

**FE-12.** Export candidate list for post-checking process, import result

#### 6.1.3 Examiner uses Web Application/Mobile Application

As a Examiner, I want to use these following features:

**FE-13.** Manage examiner account: view profile, update profile

**FE-14.** View assigned campaigns

**FE-15.** Manage test: create test, update test, view test, delete test, choose a test for a round, grade test

**FE-16.** Take appearance assessment, view appearance assessment result

**FE-17.** Take interview evaluation, view interview evaluation

**FE-18.** Manage enquiry request: View enquiry request, re-grade enquiry request

#### 6.1.4 Airline Partner uses Web Application

As a Airline Partner, I want to use these following features:

**FE-19.** Manage airline partner account: view profile, update profile

**FE-20.** Manage campaign request: create campaign request, update campaign request, view campaign request, delete campaign request

**FE-21.** Manage campaign: view campaign based on campaign request, view final result of campaign

#### 6.1.5 Admin use System Admin Web Application

As a Admin, I want to view report and manage system with the following features:

**FE-22.** Manage user account: create account, update status of account, disable account

**FE-23.** Manage dashboard

**FE-24.** Manage system configurations such as job requirement, job description

**FE-25.** Export reports/statistics

### 6.2 Limitations & Exclusions

**LE-01.** The system is primarily designed for internal deployment within the Aviation Academy, but may be extended to accommodate external airline partners seeking recruitment, training, and promotion services for cabin crew.

**LE-02.** Certain specialized requirements (e.g., medical examinations, practical training) remain outside the system’s scope and must be conducted offline.

**LE-03.** Direct in-person interviews are not fully replaced; instead, the system supports a hybrid model combining online and offline interactions.

**LE-04.** The system does not aim to provide full-scale Human Resource Management (HRM) capabilities; its scope is limited to cabin crew recruitment and promotion processes

# II. Project Management Plan

## 1. Overview

### 1.1 Scope & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| # | WBS Item | Complexity | Est. Effort |
| ***1.*** | ***Initiating*** |  |  |
| 1.1. | Collect Requirements | Medium | 14 |
| ***2.*** | ***Planning*** |  |  |
| 2.1. | Create plan document | Comples | 11 |
| 3. | ***Executing*** |  |  |
| 3.1. | **Analysis** |  |  |
| *3.1.1.* | Analysis requirement | Complex | 12 |
| 3.2. | **Design** |  |  |
| 3.2.1. | Design Candidate web application | Complex | 12 |
| 3.2.2. | Design Recruiter web application | Complex | 7 |
| 3.2.3. | Design Examiner web application | Complex | 10 |
| 3.2.4. | Design Examiner mobile application | Complex | 5 |
| 3.2.5. | Design Airline Partner web application | Complex | 5 |
| 3.2.6. | Design Admin web application | Complex | 7 |
| 3.3. | **Prototyping** |  |  |
| *3.3.1.* | Mockup | Complex | 10 |
| 3.4. | **Implementation** |  |  |
| *3.4.1.* | *Implement Base feature* |  |  |
| 3.4.1.1. | Login | Simple | 2 |
| 3.4.1.2 | Logout | Simple | 2 |
| 3.4.1.3 | Register | Simple | 3 |
| *3.4.2.* | *Implement Candidate feature* |  |  |
| 3.4.2.1. | Home Main | Medium | 4 |
| 3.4.2.2. | Search | Complex | 5 |
| 3.4.2.3. | Profile | Medium | 4 |
| 3.4.2.4. | Manage Campaign | Complex | 6 |
| 3.4.2.5. | Manage Application | Complex | 6 |
| 3.4.2.6. | Manage Result | Complex | 6 |
| 3.4.2.7. | Manage Recruitment Process | Complex | 6 |
| 3.4.2.8. | Take Test | Complex | 6 |
| *3.4.3.* | *Implement Recruiter feature* |  |  |
| 3.4.3.1. | Manage Application | Medium | 5 |
| 3.4.3.2. | Manage Campaign | Medium | 5 |
| 3.4.3.3. | Import/Export post-checking process | Complex | 6 |
| *3.4.4.* | *Implement Examiner feature* |  |  |
| 3.4.4.1. | Manage Interview Result | Complex | 6 |
| 3.4.4.2. | Manage Campaign | Medium | 5 |
| 3.4.4.3. | Manage Appearance Assessment | Complex | 6 |
| 3.4.4.4. | Manage Test | Complex | 6 |
| *3.4.5.* | *Implement Airline Partner feature* |  |  |
| 3.4.5.1. | Manage Campaign Request | Medium | 5 |
| 3.4.5.2. | Manage Campaign | Medium | 4 |
| 3.4.5.3. | Manage Result | Medium | 4 |
| *3.4.6.* | *Implement Admin feature* |  |  |
| 3.4.6.1. | Manage Account | Medium | 5 |
| 3.4.6.2. | Manage Dashboard | Complex | 6 |
| 3.4.6.3. | Manage System Configurations | Medium | 5 |
| 3.4.6.4. | Manage Reports/Statistics | Complex | 4 |
| 3.5. | **Testing** |  |  |
| *3.5.1.* | Unit Test | Complex | 10 |
| *3.5.2.* | Integration Test | Complex | 10 |
| *3.5.3.* | System Test | Complex | 10 |
| ***4.*** | ***Monitoring and Controlling*** |  |  |
| 4.1. | Control the process | Complex | 10 |
| 4.2. | Track performance a quality | Complex | 10 |
| ***5.*** | ***Closing*** |  |  |
| 5.1. | Report | Simple | 15 |
| **Total Estimated Effort (man-days): 280** | | | |

### 1.2 Project Objectives

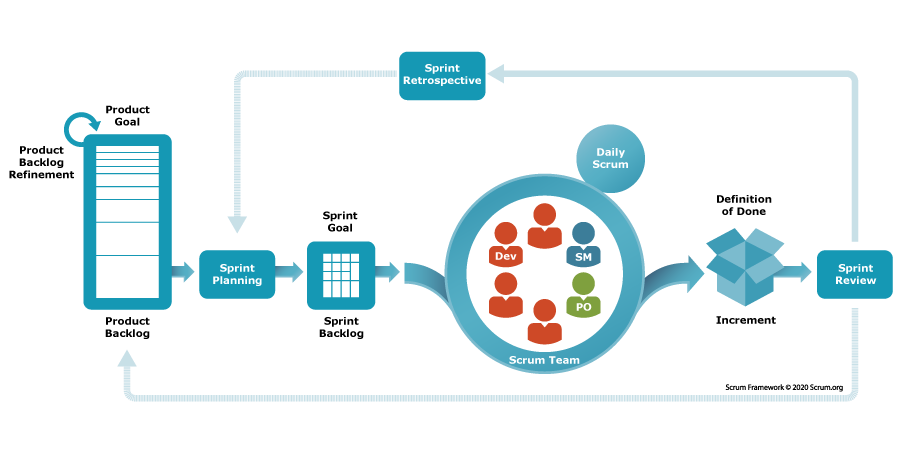
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Quality Stage** | **No. of Defects** | **% of Defect** | **Notes** |
| 1 | Reviewing | 9 | 29,03% | Focus on code quality and standards. |
| 2 | Unit Test | 6 | 19,35% | Test individual components/functions for correctness. |
| 3 | Integration Test | 4 | 12,90% | Test the interaction between integrated units. |
| 4 | System Test | 7 | 22,58% | Validate the complete system’s functionality. |
| 5 | Acceptance Test | 5 | 16,14% | Ensure the application meets user/business need. |
| **Total** | | **31** | **100%** |  |

### 1.3 Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Technical complexities or unforseen technical issues | Delays in delivery | Medium | Allocate buffer time in the project timeline. Schedule regular technical reviews. |
| 2 | Data breaches or security threats. | Damage toreputation, legal consequences | High | Implement rigorous security protocols and regular vulnerability assessments. |
| 3 | Changes in project requirements or scope. | Increased time, delays | Medium | Implement a change management protocol. Prioritize changes based on their urgency. |
| 4 | Insufficient testing leading to undetected bugs. | Reduced product quality | Medium | Allocate adequate time for thorough testing. Implement automated testing. |
| 5 | Communication gaps among team members. | Misunderstandings, delays | Low | Schedule regular team meetings. Use efficient communication tools. |
| 6 | Inaccurate estimation of time and costs. | Overrunning budget or timeline | Medium | Regularly review and adjust estimates based on actual. Use experienced estimators. |
| 7 | Low user adoption after deployment. | ROI not met | Medium | Conduct beta testing with actual users. Gather feedback and refine the product. |
| 8 | Server downtimes or crashes. | Service disruption | Low | Opt for reliable hosting services. Implement real-time monitoring. |
| 9 | Regulatory or compliance changes. | Legal implications | Low | Stay updated with local regulations. Seek legal counsel for compliance checks. |

## 2. Management Approach

### 2.1 Project Process

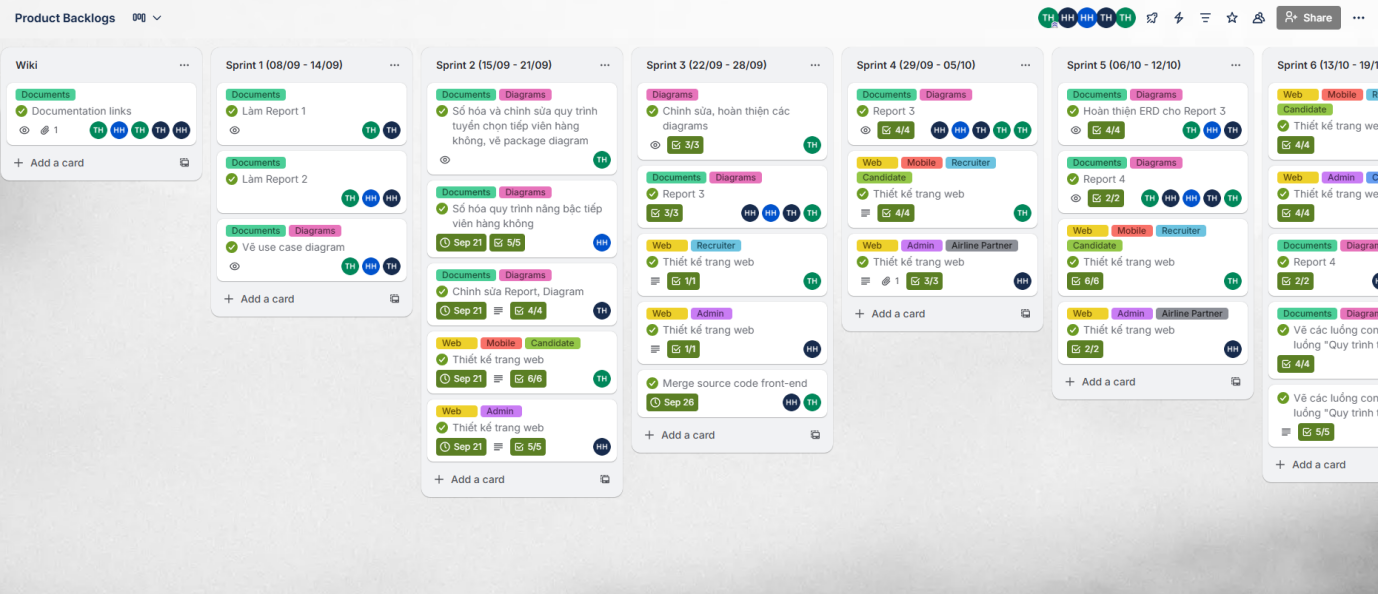


For this project, the Agile Scrum development process model is applied. The project will be divided into multiple sprints, each lasting 2 weeks. Scrum allows flexibility, continuous feedback, and incremental delivery of working software

**Process Description:**

1. **Requirement Gathering & Backlog Creation**
   * Collect requirements from the HR department and stakeholders.
   * Document user stories and prioritize them in the product backlog.
2. **Sprint Planning**
   * Select high-priority user stories for each sprint.
   * Break stories into smaller tasks and estimate effort.
3. **Design & Development**
   * Implement features iteratively.
   * Follow coding standards and best practices.
4. **Testing & Review**
   * Perform unit testing, integration testing, and peer code reviews within each sprint.
   * Ensure quality before sprint closure.
5. **Sprint Review & Retrospective**
   * Demonstrate completed features to stakeholders.
   * Collect feedback and update the backlog if needed.
   * Discussing improvements for the next sprint.
6. **Final System Integration & Acceptance**
   * After all sprints are completed, perform system testing and acceptance testing.
   * Ensure that the product meets all business and user requirements.

### 2.2 Product Backlog



### 2.3 Quality Management

|  |  |  |  |
| --- | --- | --- | --- |
| **Approach** | **Description** | **When** | **Waiver Criteria** |
| Defect Prevention | Implement best practices during development to prevent defects from arising. Includes developer training, use of coding standards, and automated tools to detect errors. | From the start of the coding phase and throughout the project life-cycle. | Mandatory |
| Reviewing | Conduct requirement, design, and code reviews to ensure adherence to standards and detect issues early. Peer reviews help identify improvement areas. | After every major deliverable (requirements, design docs, or feature implementation). | Mandatory |
| Unit Testing | Validate correctness of individual functions and modules through automated/manual unit tests. Achieve at least 80–85% code coverage. | During development of each module, before integration. | Mandatory |
| Integration Testing | Verify data flow and interaction between system components (Admin, Candidate, Recruiter, Airline Partner, Examiner apps). Automate regression tests for critical workflows. | After completion of related modules and prior to system testing. | Mandatory |
| System Testing | Validate full system functionality, performance, and security. Ensure workflows align with business requirements and user needs. | After integration testing and before acceptance testing. | Mandatory |

### 2.4 Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| .NET | NhanTT, HungTD, NguyenHNN | Week 1 - 6 days | Mandatory |
| Git, GitHub | All members | Week 1 - 2 days | Mandatory |
| ReactJs, ReactNative | HoangTN, MinhHLN, HungTD | Week 1 - 3 days | Mandatory |
| Unit test | All members | Week 2 -2 days | Mandatory |

## Project Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Deliverable** | **Due Date** | **Notes** |
| 1 | Report (Project  Introduction Document) | 22-09-2025 |  |
| 2 | Report 2 (Project Management Plan) | 22-09-2025 |  |
| 3 | Report 3 (Software Requirement Specification) | 22-10-2025 | . |
| 4 | Report 4 (Software Design Document) | 31-10-2025 |  |
| 5 | Report 5 (Test Documentation) | 05-12-2025 |  |
| 6 | Report 6 (Software User Guides) | 09-12-2025 |  |
| 7 | Report 7 (Final Project Report) | 15-12-2025 |  |

## Responsibility Assignments

*D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Responsibility** | **NhanTT** | **HungTD** | **NguyenHNN** | **HoangTN** | **MinhHLN** |
| Project Planning & Tracking | D | R | S | I | I |
| Prepare Project Introduction Document | D | R | S | I | I |
| Prepare Project Management Plan | R | D | S | I | I |
| Prepare SRS Document (Requirement Analysis) | R | S | D | S | I |
| Prepare SDD Document (System Design) | R | S | S | D | I |
| Prepare Test Documentation | R | D | S | I | S |
| Prepare Software User Guides | D | R | I | S | S |

## Project Communications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| Work in team | Team member | - Work together  - Report task  - Discuss about document and code | Weekly | - Google Meet  - Zalo  - Face to face |
| Work with supervisor | Team member + Supervisor | - Report task  - Q&A  - Give feedback and orientation | 1 - 2 day / week | - Google Meet  - Zalo  - Face to face |

## Configuration Management

### Document Management

* Use Google Drive to manage project documentation.
* Use Google Docs/Sheets to change the version of document.
* Upload diagrams, images and videos into folder named by category for easy editing and management.

### Source Code Management

* We use Git for version control and GitHub for storing and managing the source code.
* We have a few rules for using GitHub:
  + Carefully consider and be responsible for the source code pushed to the repository.
  + Take brief but thorough notes when pushing code up.
  + Review and resolve common conflicts in code.

### Tools & Infrastructures

|  |  |
| --- | --- |
| **Category** | **Tools / Infrastructure** |
| **Technology** | HTML5, SCSS, Javascript, Tailwind, ReactJS (Front-end), React Native (Mobile App), ASP.NET Core Web API (Back-end) |
| **Database** | MSSQL |
| **IDEs/Editors** | Visual Studio Code, Visual Studio |
| **Diagramming** | LucidChart |
| **Documentation** | Google Docs/Sheets/Slides |
| **Version Control** | GitHub (Source Codes), Google Drive & Google Docs/Sheets (Documents) |
| **Deployment server** | Azure Web Service, Azure Database Service, Vercel |
| **Project management** | Trello |

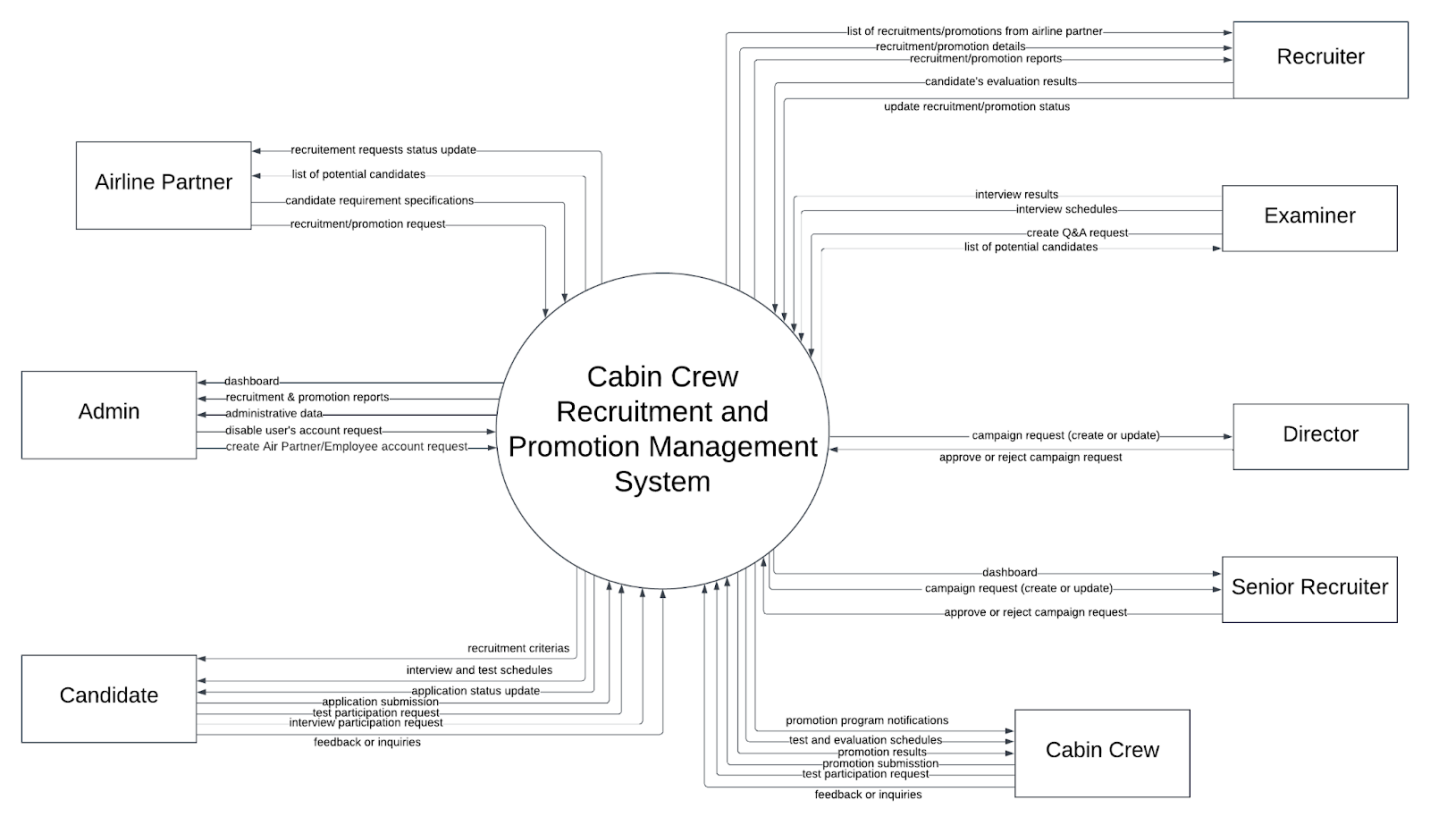
# III. Software Requirement Specification

## 1. Product Overview

The Cabin Crew Recruitment and Promotion Management System (CCRPMS) is designed to digitize and standardize the processes of recruiting and promoting cabin crew members within an Aviation Academy. The system addresses the limitations of traditional manual methods (paper-based applications, fragmented record-keeping, and non-standardized evaluations) by providing a comprehensive online platform for candidates, cabin crew, recruiters, examiners, and airline partners.

Key objectives of the system:

* Streamline recruitment and promotion processes.
* Enhance transparency for candidates and cabin crews.
* Reduce errors and administrative workload for HR staff.
* Provide data-driven insigts for management and airline partners.
* Ensure compliance with aviation industry standards and digital transformation initiatives.



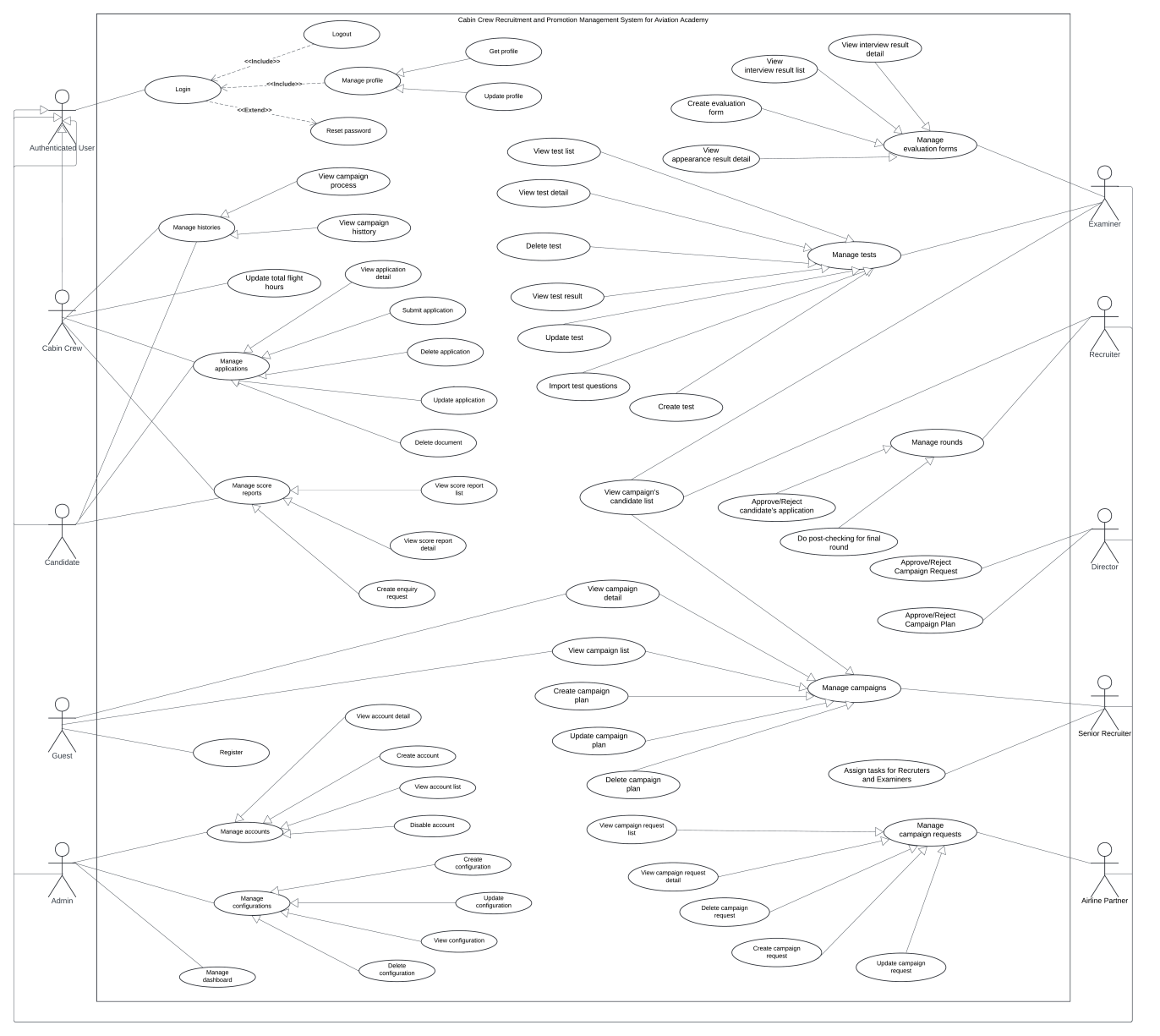
## 2. User Requirements

### 2.1 Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Administrator | Manages the overall system, assign roles, maintain security and database integrity. |
| 2 | Recruiter | Manages application, schedules interviews/exams, and communicates results to candidates |
| 3 | Examiner | Evaluates candidates or cabin crews during interviews, tests, and appearance assessment, and submits scores directly into the system. |
| 4 | Candidate | External applicants who register, submit applications, take exams, attend interviews, and track their recruitment process |
| 5 | Cabin Crew | Current employees are eligible for promotion; they register for promotion exams, attend assessments, and receive results. |
| 6 | Airline Partner | External airline representatives who may view reports, collaborate on joint recruitment or promotion programs, and provide input on required standard. |
| 7 | Director | Reviews and approves campaign requests submitted by Airline Partners or Senior Recruiter. Has the authority to approve or reject campaigns with remarks. |
| 8 | Senior Recruiter | Creates detailed recruitment plans, and forwards them to the Director for approval. After approval, assigns employees to execute the campaigns. |

### 2.2 Use cases

#### 2.2.1 Diagram



#### 2.2.2 Descriptions

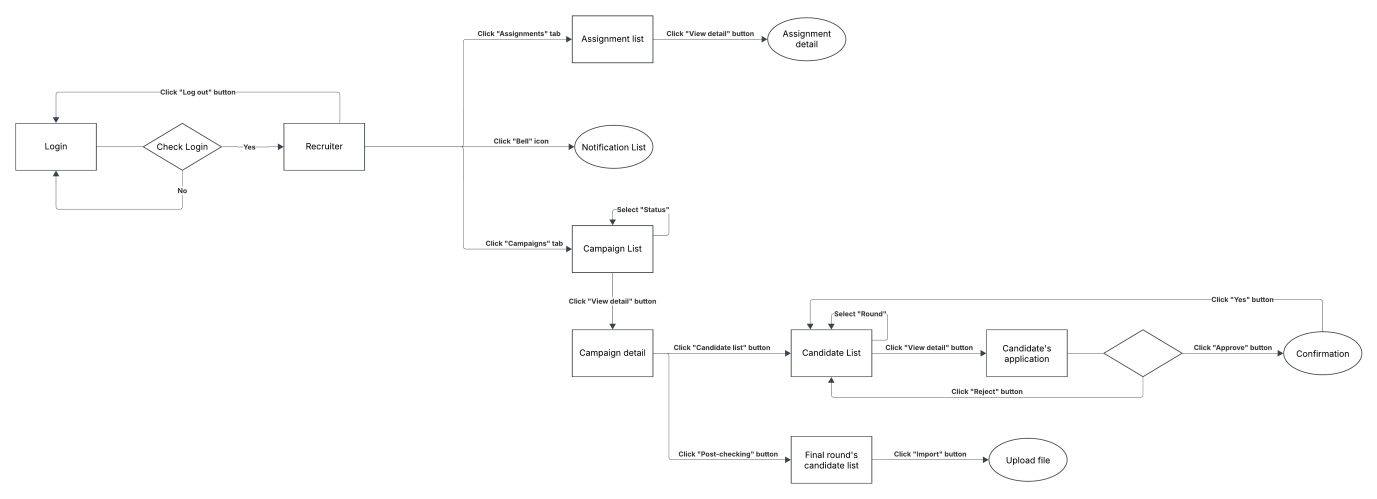
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description** |
| UC-01 | Login | Authenticated User | This feature allows user to log into the system |
| UC-02 | Logout | Authenticated User | This feature allows user to log out of the system |
| UC-03 | Reset Password | Authenticated User | This feature allows user to reset their password |
| UC-04 | Get profile | Authenticated User | This feature allows user to view their personal profile |
| UC-05 | Update profile | Authenticated User | This feature allows user to update their profile |
| UC-06 | View promotional announcements | Cabin Crew | This features allow cabin crew to view all the promotional announcements |
| UC-07 | Create promotional application | Cabin Crew | This features allows cabin crew to create their promotional application |
| UC-08 | Update promotional application | Cabin Crew | This feature allows cabin crew to update their promotional application |
| UC-09 | View promotional application criterias | Cabin Crew | This feature allows cabin crew to view the promotional application’s criterias |
| UC-10 | View promotional history | Cabin Crew | This feature allows cabin crew to view their promotional history |
| UC-11 | Delete promotional application | Cabin Crew | This feature allows cabin crew to delete their promotional application |
| UC-12 | View promotional test result | Cabin Crew | This feature allows cabin crew to view their promotional test result |
| UC-13 | Submit promotional test | Cabin Crew | This feature allows cabin crew to submit their promotional test |
| UC-14 | Register | Guest | This feature allows guest to register an account |
| UC-15 | View test result | Candidate | This feature allows candidate to view their test result |
| UC-16 | Submit English test | Candidate | This feature allows candidate to submit their English test |
| UC-17 | View test schedule | Candidate | This feature allows candidate to view their test schedule |
| UC-18 | Delete feedback | Candidate | This feature allows candidate to delete their feedback |
| UC-19 | View feedback list | Candidate | This feature allows candidate to view the feedback list |
| UC-20 | Update feedback | Candidate | This feature allows candidate to update their feedback |
| UC-21 | Create feedback | Candidate | This feature allows candidate to create their feedback |
| UC-22 | View application details | Candidate | This feature allows candidate to view their submitted application |
| UC-23 | Submit application | Candidate | This feature allows candidate to submit their application |
| UC-24 | Delete application | Candidate | This feature allows candidate to delete their application |
| UC-25 | Update application | Candidate | This feature allows candidate to update their application |
| UC-26 | Submit document | Candidate | Candidate uploads required documents (photo, CV, certificates) for a campaign. The system validates and saves them. |
| UC-27 | Update document | Candidate | Candidate replaces previously uploaded files before the campaign deadline. System updates to the latest version. |
| UC-28 | Delete document | Candidate | Candidate deletes uploaded files if the campaign is still open. System confirms before removal. |
| UC-29 | Manage dashboard | Admin | This feature allows admin to manage the system’s dashboard |
| UC-30 | Manage interview schedules | Admin | This feature allows admin to schedule candidate/cabin crew interview |
| UC-31 | View campaign detail | Admin | This feature allows admin to view a campaign detail |
| UC-32 | View campaign list | Admin | This feature allows admin to view the list of campaigns |
| UC-33 | Delete campaign | Admin | This feature allows admin to delete a specific campaign |
| UC-34 | Update campaign | Admin | This feature allows admin to update a specific campaign |
| UC-35 | Create campaign | Admin | This feature allows admin to create a campaign |
| UC-36 | Disable account | Admin | This feature allows admin to disable an account |
| UC-37 | View account list | Admin | This feature allows admin to view the list of accounts |
| UC-38 | Create account | Admin | This feature allows admin to create an account |
| UC-39 | View account detail | Admin | This feature allows admin to view a specific account’s profile |
| UC-40 | Submit recruitment requirements | Airline partner | This feature allows airline partners to submit a recruitment requirements |
| UC-41 | View collaboration reports | Airline partner | This feature allows airline partners to view collaboration reports |
| UC-42 | View candidate list | Airline partner/Recruiter | This feature allows airline partner/recruiter to view the list of candidates |
| UC-43 | Create evaluation test | Recruiter | This feature allows recruiter to create an evaluation test |
| UC-44 | View Q&A detail | Recruiter/Candidate/Cabin Crew | This feature allows recruiter/candidate/cabin crew to view Q&A details |
| UC-45 | Delete Q&A | Recruiter | This feature allows recruiter to delete a specific Q&A |
| UC-46 | Update Q&A | Recruiter | This feature allows recruiter to update a specific Q&A |
| UC-47 | Create Q&A | Recruiter | This feature allows recruiter to create a Q&A |
| UC-48 | View cabin crew list | Recruiter | This feature allows recruiter to view the list of cabin crews |
| UC-49 | Manage appearance assessment | Recruiter | This feature allows recruiter to manage candidate’s appearance assessment |
| UC-50 | Manage recruitment reports | Recruiter | This feature allows recruiter to manage recruitment reports |
| UC-51 | Store/Manage data | System Handler | System stores and manages data |
| UC-52 | Send notifications | System Handler | System sends notifications to all users |
| UC-53 | Validation applications | System Handler | System validates all the requests sent by users |
| UC-54 | Generate reports | System Handler | System generates reports based on the stored data |
| UC-55 | Review Campaign Requests | Senior Recruiter | Senior Recruiter reviews campaign requests submitted by Recruiters or Airline Partners, checks feasibility, and prepares summary for Director. |
| UC-56 | Approve Campaign | Director | Director approves campaign requests or recruitment plans proposed by the Senior Recruiter or HR. |
| UC-57 | Reject Campaign (with note) | Director | The director rejects a campaign request, providing reasons or revision notes. |
| UC-58 | Send task to Recruiter and Examiner | Senior Recruiter | After Director approval, Senior Recruiter assigns Recruiters to handle each campaign round. |
| UC-59 | Send Campaign Request to HR | Airline Partner | Airline Partner submits campaign or recruitment requests to be reviewed by the Senior Recruiter. |
| UC-60 | Submit Crew Experience Documents | Airline Partner | Airline Partner uploads flight experience or candidate performance data for Cabin Crew records |
| UC-61 | Manage Flight Experience | Recruiter / Admin | Allows authorized users to view, verify, and update flight experience entries linked to Cabin Crew profiles. |
| UC-62 | Create Campaign Round | Recruiter | Allows Recruiters to create campaign sub-rounds after main campaign approval. |
| UC-63 | Manage Campaign Approval Flow | System Handler | Automates notifications and status updates through approval stages (Submitted → Reviewed → Approved/Rejected). |

## 3. Functional Requirements

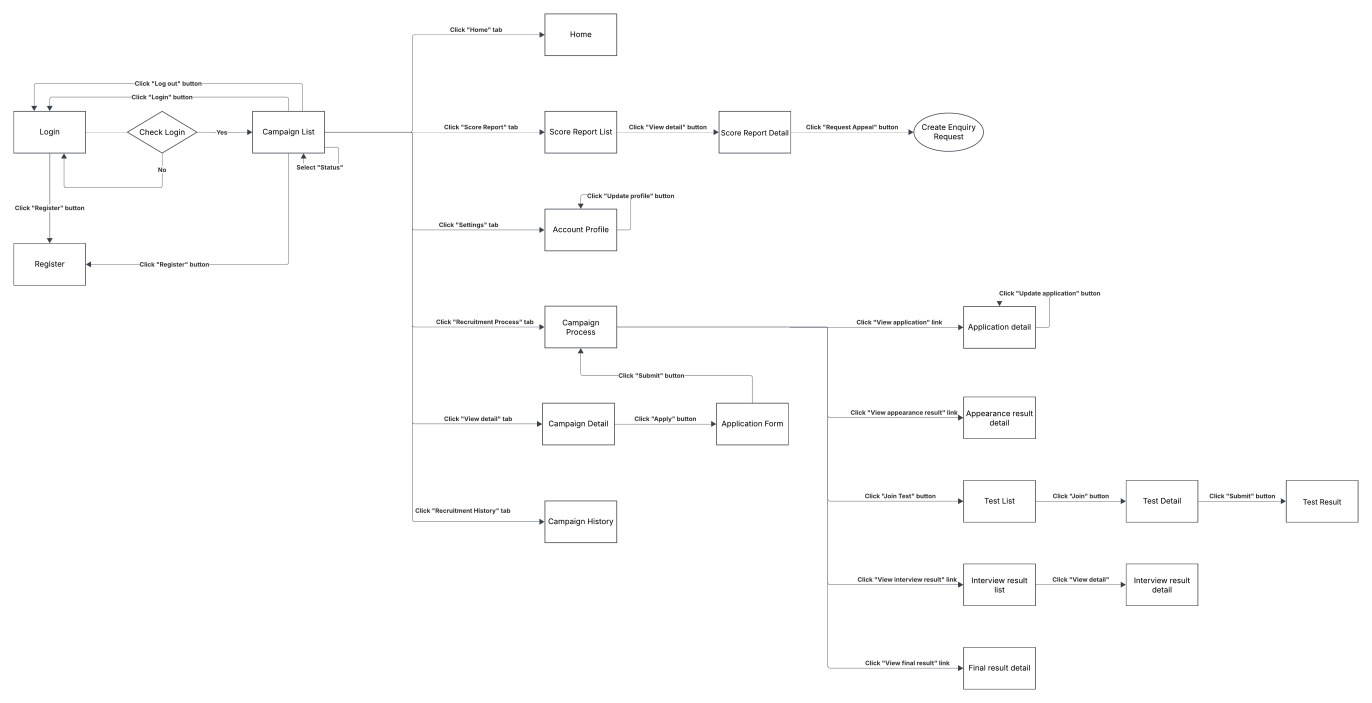
### 3.1 System Functional Overview

#### 3.1.1 Screens Flow

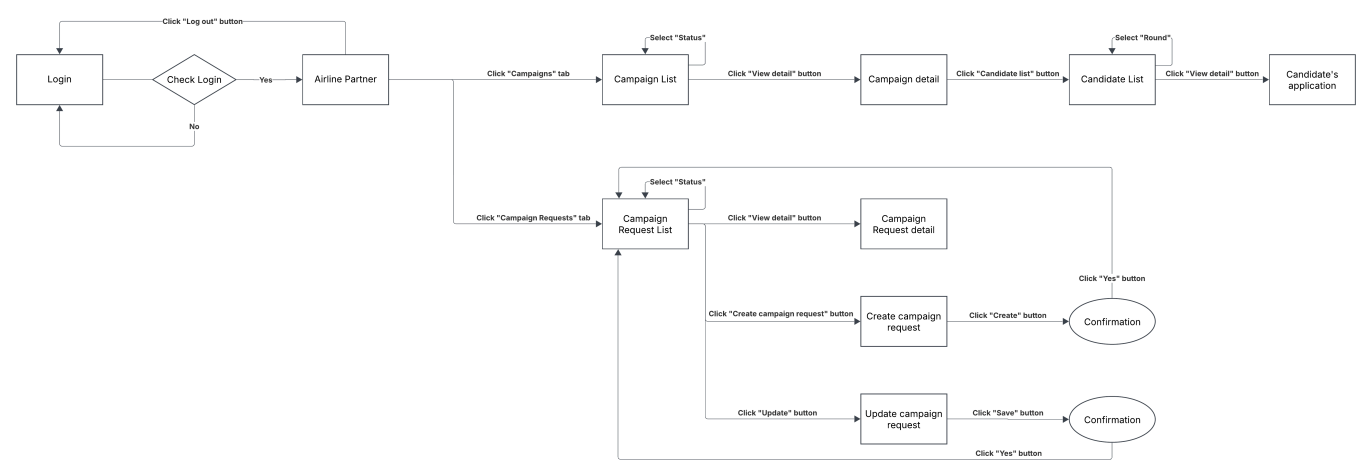
##### 3.1.1.1 Recruiter Web App



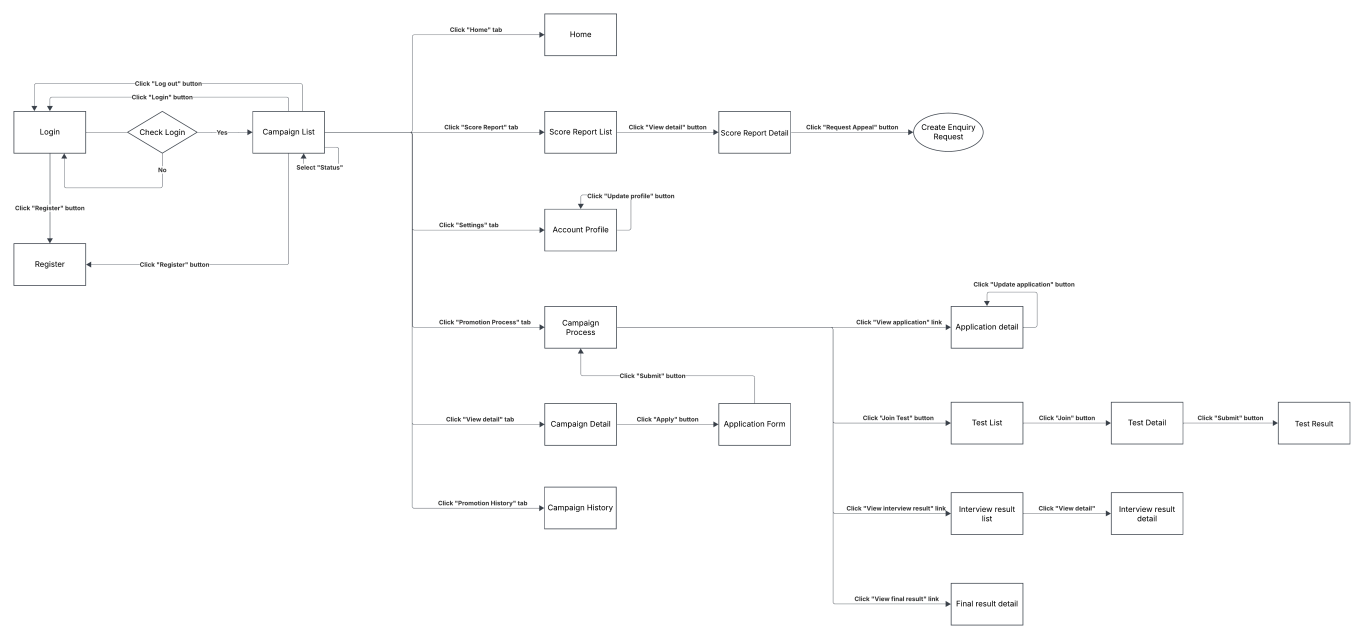
##### 3.1.1.2 Candidate Web App



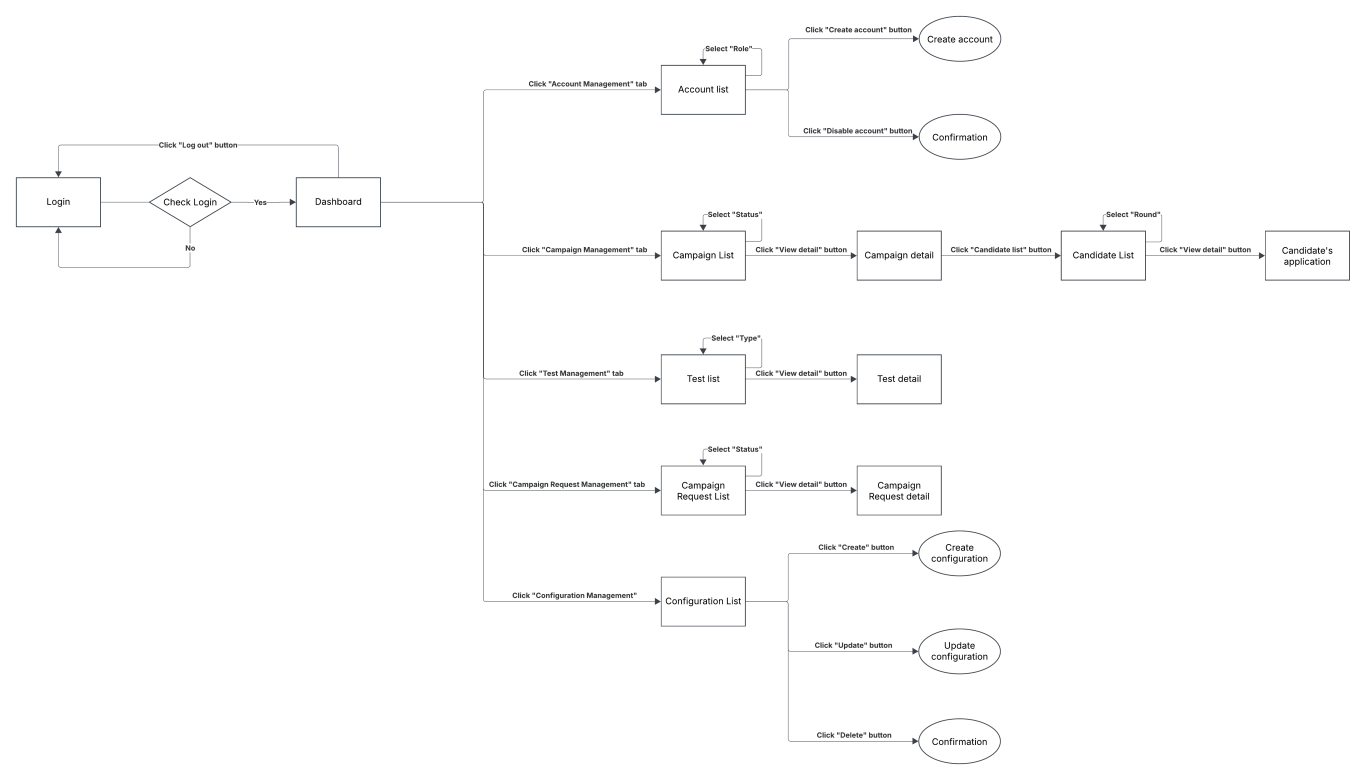
##### 3.1.1.3 Airline Partner Web App



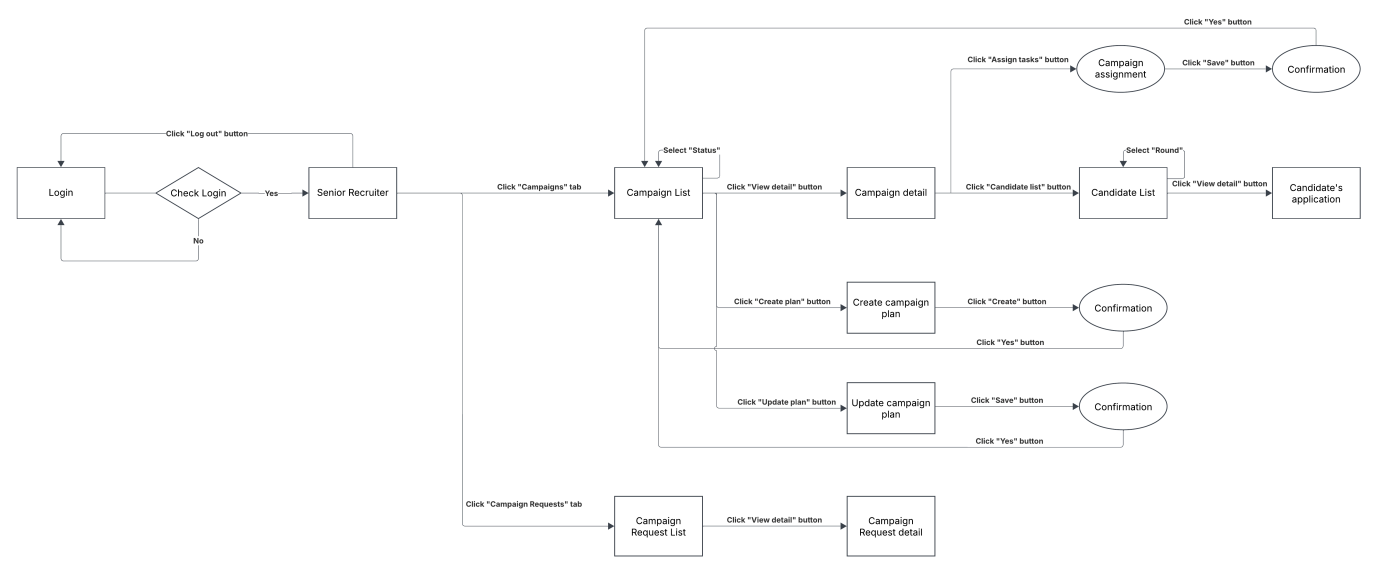
##### 3.1.1.4 Cabin Crew Web App



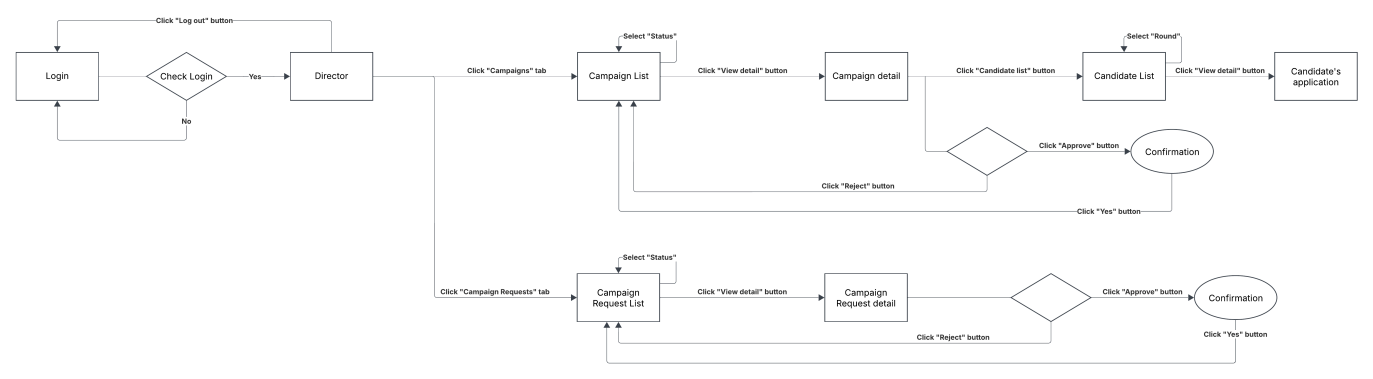
##### 3.1.1.5 Admin Web App



##### 3.1.1.6 Senior Recruiter Web App

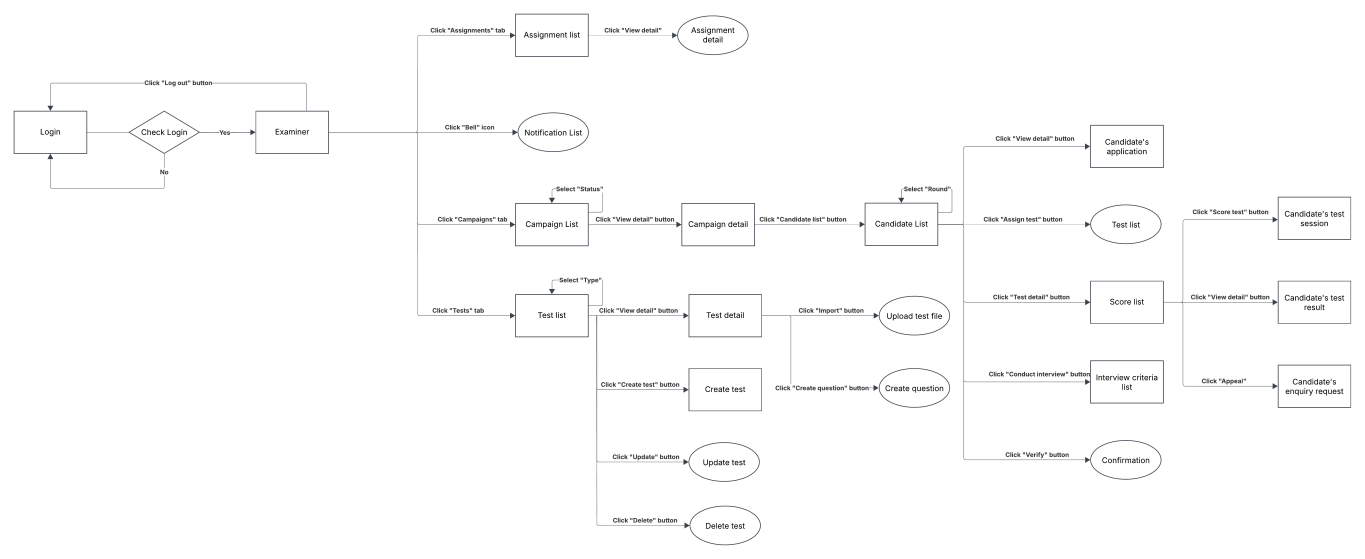


##### 3.1.1.7 Director Web App

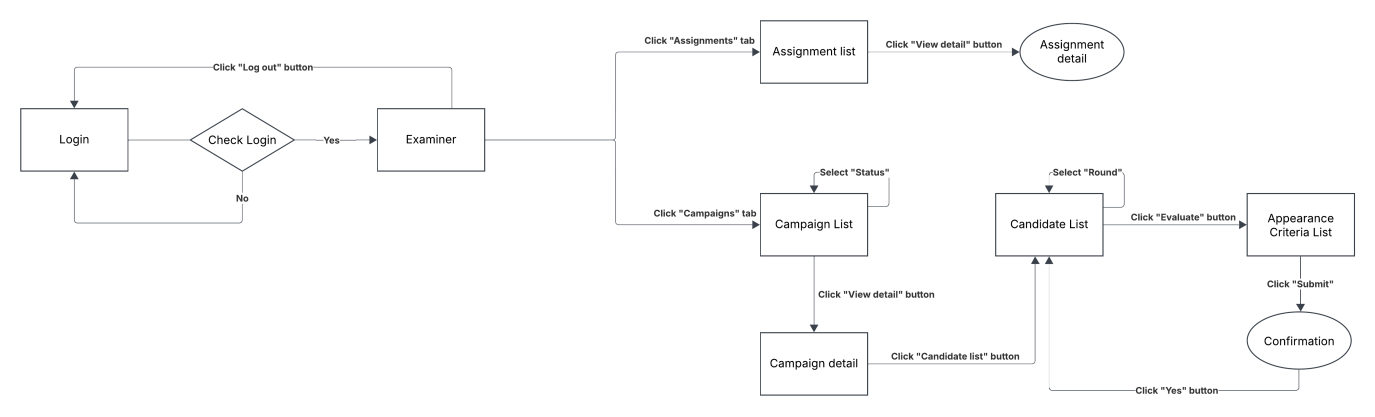


##### 3.1.1.8 Examiner

###### 3.1.1.8.1 Web app



###### 3.1.1.8.2 Mobile app



#### 3.1.2 Screens Description

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | General Access | Home | Accessible to all roles (Administrator, Recruiter, Examiner, Candidate, Cabin Crew, Airline Partner, Director, Senior Recruiter). Displays general announcements, navigation menu, and system overview. |
| 2 | User Management | Login | Shared screen for all user roles to securely log into the system using credentials and two-factor authentication. |
| 3 | User Management | Register | Allows new users (Candidate, Cabin Crew, Airline Partner) to register accounts with email/phone verification. |
| 4 | Dashboard | Dashboard | Dashboard for Administrator only. Displays system statistics, campaign summaries, recruitment progress, and active user counts. |
| 5 | User Management | Account profile | Accessible to Recruiter, Candidate, Cabin Crew, and Airline Partner to view and update personal details and contact information. |
| 6 | Campaign Management | Campaign list | Visible to Admin, Recruiter, Candidate, Cabin Crew, Airline Partner, Director. Displays a list of all campaigns (recruitment and promotion) with filtering and search functions. |
| 7 | Campaign Management | Campaign detail | Used by Recruiter, Candidate, Cabin Crew, Airline Partner, and Director to view campaign-specific information, including schedule, requirements, and progress. |
| 8 | Campaign Request Management | Create campaign request | Screen for Airline Partner to submit new recruitment or collaboration requests to the Academy, including campaign goals, description, and target number. |
| 9 | Campaign Management | Create campaign plan | Used by Administrator to create new recruitment or promotion campaigns, including campaign name, type, timeline, and responsible personnel. |
| 10 | Campaign Management | Update campaign plan |  |
| 11 | Application Management | Application form | Screen for Candidate and Cabin Crew to submit applications for recruitment or promotion campaigns, attach required documents, and track submission status. |
| 12 | Application Management | Application detail | Allows Candidate and Cabin Crew to view submitted applications, document list, and current approval/review status. |
| 13 | History Management | Campaign history | Candidate screen to view history of recruitment applications, scores, and final results from previous campaigns. |
| 14 | History Management | Campaign progress | Cabin Crew screen to view their promotion participation history, exam results, and approval status. |
| 15 | Round Management | Candidate list | Administrator’s screen to view all candidate accounts, manage access rights, or deactivate users. |
| 16 | Round Management | Candidate application | Recruiter screen to review submitted applications, verify eligibility, and approve or reject candidates for the next stage. |
| 17 | Account Management | Account list | Admin screen to view all system accounts, their roles, and current status (active/inactive). |
| 18 | Account Management | Create Account | Admin screen for creating new user accounts under specific roles (Recruiter, Examiner, Cabin Crew, etc.) with appropriate permissions. |
| 19 | Test Management | Test list | Screen for Candidate to take English proficiency tests online. Accessible to Examiner for monitoring and scoring. |
| 20 | Test Management | Test Detail | Screen for Examiner to see the detail of a specific test |
| 21 | Test Management | Test Result | Screen for Candidate/Cabin Crew to see their result after taking a test |
| 22 | Test Management | Create test | Recruiter screen for creating new online or written tests, defining question pools, difficulty levels, and duration. |
| 23 | Test Management | Score report list | Screen show the score report for each test Candidate/Cabin Crew has done |
| 24 | Test Management | Score report detail | Screen show the detail score report of how Examiner evaluated the test |
| 25 | Interview Management | Interview result list | Interface for Recruiters and Examiners to schedule, conduct, and manage interview sessions with selected candidates or cabin crew. |
| 26 | Interview Management | Interview result detail | Screen show the interview result detail with all the criterias Examiner has evaluated |
| 27 | Appearance  Management | Appearance criteria list | Screen for Examiners to input physical measurement data (height, weight, BMI) and assign scores for candidates’ physical requirements. |
| 28 | Appearance  Management | Appearance result detail | Screen for Examiners to evaluate grooming, posture, and appearance standards of candidates or crew members. |
| 29 | History Management | Final result detail | Shared screen for Admin, Recruiter, Examiner, Candidate, Cabin Crew to view final test and interview results, including pass/fail status and comments. |
| 30 | Task Management | Assignment list | Screen for Recruiter to view assigned tasks and campaign responsibilities within the recruitment workflow. |
| 31 | Task Management | Assignment detail | Displays details of each task assigned to Recruiters, including campaign, deadlines, and linked applicants. |
| 32 | Task Management | Campaign assignment | Screen for Senior Recruiter to assign tasks (screening, evaluation, communication) to Recruiters or Examiners. |
| 33 | Configuration Management | Configuration list | Screen for Admin to configure some general properties with the app |

#### 3.1.3 Screens Authorization

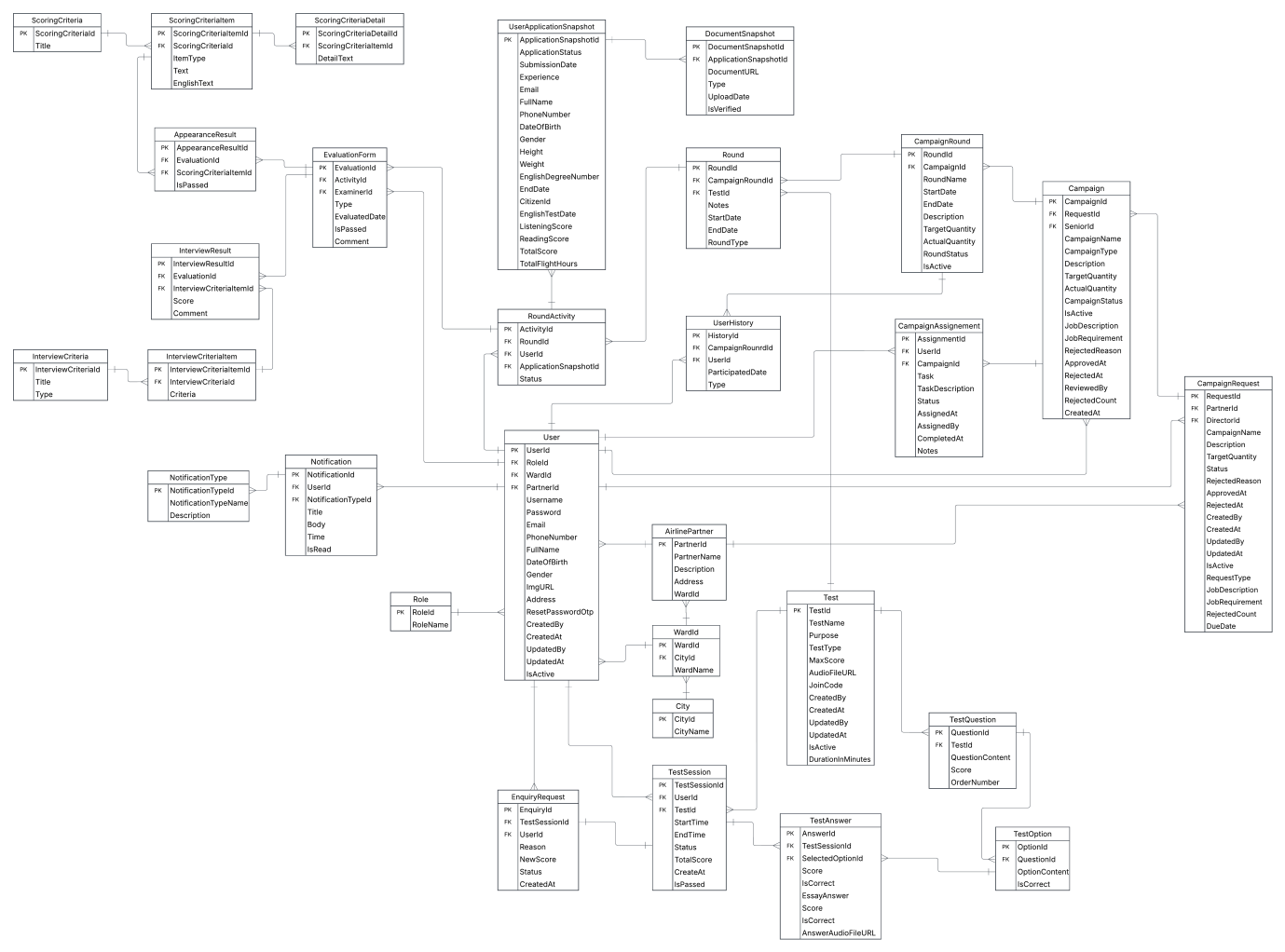
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Screen** | **Admin** | **Recruiter** | **Examiner** | **Candidate** | **Cabin Crew** | **Airline Partner** | **Director** | **Senior Recruiter** |
| Home | - | - | - | X | X | - | - | - |
| Login | X | X | X | X | X | X | X | X |
| Register | X | X | X | X | X | X | X | X |
| Dashboard | X | - | - | - | - | - | - | - |
| Account profile | - | X | - | X | X | X | - | - |
| Campaign list | X | X | X | X | X | X | X | X |
| Campaign detail | X | X | X | X | X | X | X | X |
| Create campaign request | - | - | - | - | - | X | - | - |
| Create campaign plan | - | - | - | - | - | - | - | X |
| Update campaign plan | - | - | - | - | - | - | - | X |
| Application form | - | - | - | X | X | - | - | - |
| Application detail | - | - | - | X | X | - | - | - |
| Campaign history | - | - | - | X | X | - | - | - |
| Campaign progress | - | - | - | X | X | - | - | - |
| Candidate list | X | X | X | - | - | X | X | X |
| Candidate application | X | X | X | - | - | X | X | X |
| Account list | X | - | - | - | - | - | - | - |
| Create Account | X | - | - | - | - | - | - | - |
| Test list | X | - | X | X | X | - | - | - |
| Test Detail | X | - | X | X | X | - | - | - |
| Test Result | - | - | - | X | X | - | - | - |
| Create test | - | - | X | - | - | - | - | - |
| Score report list | - | - | X | X | X | - | - | - |
| Score report detail | - | - | X | X | X | - | - | - |
| Interview result list | - | - | X | X | X | - | - | - |
| Interview result detail | - | - | X | X | X | - | - | - |
| Appearance criteria list | - | - | X | X | X | - | - | - |
| Appearance result detail | - | - | X | X | X | - | - | - |
| Final result detail | X | X | X | X | X | X | X | X |
| Assignment list | - | X | X | - | - | - | - | X |
| Assignment detail | - | X | X | - | - | - | - | - |
| Campaign assignment | - | - | - | - | - | - | - | X |
| Configuration list | X | - | - | - | - | - | - | - |

#### 3.1.4 Non-screens Functions

|  |  |  |
| --- | --- | --- |
| **#** | **System Function** | **Description** |
| 1 | Send Mails to User | Sends mails to users when there are updates about  the information of Application, Appearance Assessment, Test, Interview and Final. |
| 2 | Send Notifications to User | Sends notifications to users when there are updates about  the information of Campaign Request, Campaign, Campaign Assignment. |
| 3 | Add User to next Round | Automatically add user to the next round if he/she passes the current Round. |
| 4 | Update User History Status to “Completed” | Update “Ongoing” User History to “Completed” status when he/she passes all the Rounds. |
| 5 | Update User History Status to “Not Completed” | Update “Ongoing” User History to “Not Completed” status when he/she fails a single Round. |
| 6 | Update Campaign Status to “Ongoing” | Update “Upcoming” Campaign to “Ongoing” status when it has reached its Start Date. This service runs every 5 minutes. |
| 7 | Update Campaign Round Status to “Ongoing” | Update “Upcoming” Campaign Round to “Ongoing” status when it has reached its Start Date. This service runs every 5 minutes. |

#### 3.1.5 Entity Relationship Diagram

##### 3.1.5.1 Diagram



##### 3.1.5.2 Entities Description

|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | User | Central entity representing all system users, including Administrator, Recruiter, Examiner, Candidate, Cabin Crew, and Airline Partner. Stores login credentials, personal information, contact details, and links to roles and notifications. |
| 2 | Role | Defines each user’s access level and permissions. Connected to User through a one-to-many relationship. Examples: Admin, Recruiter, Examiner, Candidate, Cabin Crew, Airline Partner. |
| 3 | AirlinePartner | Contains information about external airline organizations collaborating with the system. Used to associate internal crew data and uploaded files (e.g., KPI, training logs) with specific partners. |
| 4 | Campaign | Represents a recruitment or promotion program created by Senior Recruiters. Includes details such as campaign name, type, timeline, description, and current status. |
| 5 | CampaignRound | Defines a sub-phase or batch within a campaign (e.g., “March 2025 Round”). Each campaign may have multiple campaign rounds, allowing multiple recruitment or promotion sessions under one campaign. |
| 6 | Round | Defines a round with a campaign round. Each campaign round may have multiple rounds, allowing multiple recruitment or promotion sessions under one campaign. |
| 7 | RoundActivity | Tracks to see how candidates/cabin crews are progressing during each round |
| 8 | CampaignAssignment | Store recruiters/examiners’ task assigned by the senior recruiter for one or more campaigns |
| 9 | CampaignRequest | Define a campaign request created by an Airline Partner in order to meet their desires |
| 10 | EvaluationForm | A form contains all information for a Appearance/Interview round |
| 11 | ScoringCriteria | Store the criterias used for appearance asessment round. |
| 12 | ScoringCriteriaItem | A sub-criteria belongs to a parent criteria |
| 13 | ScoringCriteriaDetail | Holds the detail that define how each item of a scoring criteria will be used to evaluate candidates/cabin crews |
| 14 | AppearanceResult | Stores the appearance results from examiners, including criteria scores, comments, and final remarks by examiners. |
| 15 | InterviewResult | Stores the interview results from panel interviews, including criteria scores, comments, and final remarks by examiners. |
| 16 | InterviewCriteria | Store the criterias used for interview round. |
| 17 | InterviewCriteriaItem | Store the detail of each criteria used for interview round. |
| 18 | Notification | Contains notifications sent to users (system alerts, campaign updates, schedule changes, or results). Each record includes title, body, time, and read status. |
| 19 | NotificationType | Defines types of notifications (e.g., System, Recruitment, Result Announcement) and their descriptions for system categorization |
| 20 | Test | Contains details about online or written exams, including test name, type, language, purpose, time limit, and the creator (Recruiter or Examiner). |
| 21 | TestQuestion | Holds the list of questions assigned to a specific test. Each question is linked to one or more answers. |
| 22 | TestOption | Stores possible answers for each question, including a flag indicating the correct answer. |
| 23 | TestSession | Stores results of users who completed a test, including score, status (Pass/Fail), and submission timestamp. Links users to the test they took. |
| 24 | TestAnswer | Captures the answer submitted by candidates/cabin crews and based on it to decide if they has passed or not. |
| 25 | EnquiryRequest | Holds an enquiry request when candidates/cabin crews choose to appeal their test result. |
| 26 | City | Lists available cities used for address references in user profiles or partner locations. |
| 27 | Ward | Contains sub-level administrative regions linked to City for more precise address mapping. |
| 28 | UserApplicationSnapshot | Contains all the information candidates/cabin crews submitted when they apply for a campaign round. |
| 29 | DocumentSnapshot | Stores details about the required documents uploaded by candidates (e.g., CV, certificates), including file name and location (URL). |
| 30 | UserHistory | Captures all records when candidates/cabin crews choose to apply for any campaigns. |

### 3.2 Functional Requirements

#### 3.2.1 Recruiter functional requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Actor** | **Function Name** | **Data Requirements** | **Data Validation** | **Business Rules** |
| 1 | Recruiter | Manage Screening Rounds | ApplicationSnapshotId, Height, Weight, EnglishDegreeNumber, CitizenId, EnglishTestDate, ListeningScore, ReadingScore, TotalScore, DocumentSnapshots | ApplicationSnapshotId: Integer (Primary Key)  Height, Weight, ListeningScore, ReadingScore, TotalScore: Integer (≥ 0)  EnglishTestDate: DateOnly  EnglishDegreeNumber: String (Max 50)  CitizenId: String (Max 20)  DocumentSnapshots: Collection of DocumentSnapshot records |  |
| 2 | Recruiter | Manage Final Rounds | ApplicationSnapshotId, FullName, DateOfBirth, CitizenId, EnglishTestDate, EnglishDegreeNumber, ListeningScore, ReadingScore, TotalScore | ApplicationSnapshotId: Intege (Primary Key)  FullName, EnglishDegreeNumber: String (Max 50)  DateOfBirth, EnglishTestDate: DateOnly  ListeningScore, ReadingScore, TotalScore: Integer (≥ 0) |  |

#### 3.2.2 Candidate functional requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Actor** | **Function Name** | **Data Requirements** | **Data Validation** | **Business Rules** |
| 1 | Candidate | Apply To Recruitment Campaign Round | FullName, Email, PhoneNumber, Gender, DateOfBirth, CitizenId, EnglishTestDate, EnglishDegreeNumber, ListeningScore, ReadingScore, TotalScore, SubmisstionDate, EndDate,  DocumentSnapshots | FullName, Email, EnglishDegreeNumber: String (Max 50)  PhoneNumeber: String (Max 11)  Gender: Enum (Male, Female, Others)  DateOfBirth, EndDate, EnglishTestDate: DateOnLy  SubmissionDate: DateTime  ListeningScore, ReadingScore, TotalScore: Integer(≥ 0)  DocumentSnapshots: Collection of DocumentSnapshots records | BR-20 |
| 2 | Candidate | Manage Recruitment Process | HistoryId, ParticipatedDate, Type, CampaignRoundId, UserId, RoundActivities | HistoryId: Integer (Primary Key)  ParticipatedDate: DateOnly  Type: Enum (Ongoing, Completed, NotCompleted)  UserId: Integer (Foreign Key)  CampaignRoundId: Integer (Foreign Key)  RoundActivities: Collection of RoundActivity records |  |
| 3 | Candidate | Take English Listening Test | TestId, JoinCode, StartTime, EndTime, QuestionId, SelectedOptionId | TestId: Integer (Foreign Key)  JoinCode: String (Max 10)  StartTime, EndTime: DateTime  QuestionId: Integer (Foreign Key)  SelectedOptionId: Integer (Foreign Key) |  |
| 4 | Candidate | Take English Speaking Test | TestId, JoinCode, StartTime, EndTime, QuestionIds, AudioFiles | TestId: Integer (Foreign Key)  JoinCode: String (Max 10)  StartTime, EndTime: DateTime  QuestionIds: Collection of Integers (Foreign Keys)  AudioFiles: Collection of Audio files |  |

#### 3.2.3 Airline Partner functional requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Actor** | **Function Name** | **Data Requirements** | **Data Validation** | **Business Rules** |
| 1 | Airline Partner | Manage Campaign Requests | RequestId, CampaignName, Desciption, JobDecription, JobRequirement, TargerQuantity  Status, RequestType, DueDate, PartnerId, DirectorId, Campaigns | RequestId: Integer (Primary Key)  CampaignName, Description: String (Max 50)  JobDescription, JobRequirement: String (Max 250)  TargetQuantity: Integer (≥ 0)  Status: Enum (Pending, Approved, Rejected, Cancelled)  RequestType: Enum (Recruitment, Promotion)  DueDate: DateOnly  PartnerId: Integer (Foreign Key)  DirectorId: Integer (Foreign Key)  Campaigns: Collection of Campaign records | BR-20 |
| 2 | Airline Partner | Manage Filight Hours Confimation Round | ApplicationSnapshotId, CitizenId, FullName, Experience, TotalFlightHours | ApplicationSnapshotId: Integer (Primary Key)  Citizenid: String (Max 20)  Experience: String (Max 50)  TotalFlightHours: Integer (≥ 0) |  |

#### 3.2.4 Cabin Crew functional requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Actor** | **Function Name** | **Data Requirements** | **Data Validation** | **Business Rules** |
| 1 | Cabin Crew | Apply To Promotion Campaign Round | FullName, Email, PhoneNumber, Gender, DateOfBirth, CitizenId, EnglishTestDate, EnglishDegreeNumber, ListeningScore, ReadingScore, TotalScore, SubmisstionDate, EndDate,  DocumentSnapshots | FullName, Email, EnglishDegreeNumber: String (Max 50)  PhoneNumeber: String (Max 11)  Gender: Enum (Male, Female, Others)  DateOfBirth, EndDate, EnglishTestDate: DateOnLy  SubmissionDate: DateTime  ListeningScore, ReadingScore, TotalScore: Integer(≥ 0)  DocumentSnapshots: Collection of DocumentSnapshots records | BR-20 |
| 2 | Cabin Crew | Manage Promotion Process | HistoryId, ParticipatedDate, Type, CampaignRoundId, UserId, RoundActivities | HistoryId: Integer (Primary Key)  ParticipatedDate: DateOnly  Type: Enum (Ongoing, Completed, NotCompleted)  UserId: Integer (Foreign Key)  CampaignRoundId: Integer (Foreign Key)  RoundActivities: Collection of RoundActivity records |  |
| 3 | Cabin Crew | Take Practical Test | TestId, JoinCode, StartTime, EndTime, QuestionId, SelectedOptionId | TestId: Integer (Foreign Key)  JoinCode: String (Max 10)  StartTime, EndTime: DateTime  QuestionId: Integer (Foreign Key)  SelectedOptionId: Integer (Foreign Key) |  |

#### 3.2.5 Admin functional requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Actor** | **Function Name** | **Data Requirements** | **Data Validation** | **Business Rules** |
| 1 | Admin | Manage Accounts | UserName, FullName, PhoneNumber, Email, DateOfBirth, RoleId | UserName: String (Max 12)  FullName, Email: String (Max 50)  PhoneNumber: String (Max 11)  DateOfBirth: DateOnly  RoleId: Integer (Foreign Key) | BR-20 |
| 2 | Admin | View Dashboard Analytics | Campaigns of Airline Partner, Number of campaigns by moths, Number of candidate/cabin crew who has applied, etc. | CampaignType: Enum (Recruitment, Promotion)  CampaignStatus: Enum (Draft, Pending, Approved, Rejected, Cancelled, Ongoing, Upcoming, Ended)  TestType: Enum (EnglishListening, EnglishSpeaking, Practical)  RoundType: Enum (Screening, Appearance, FlightHoursConfirmation, EnglishListeningTest, EnglishSpeakingTest, Interview, Final) |  |
| 3 | Admin | Manage configurations | JobDescription, JobRequirement,  NumberOfPeople | JobDescription: String (Max)  JobRequirement: String (Max)  NumberOfPeople: Integer (≥ 0) |  |

#### 3.2.6 Examiner functional requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Actor** | **Function Name** | **Data Requirements** | **Data Validation** | **Business Rules** |
| 1 | Examiner | Manage Appearance Rounds | EvaluationId, EvaluationType, EvaluatedAt, Comment, IsPassed, ActivityId, ExaminerId, AppearanceResults, ScoringCriterias | EvaluationId: Integer (Primary Key)  EvaluationType: Enum (Appearance, Interview)  Comment: String (Max 250)  IsPassed: bool  ActivityId: Integer (Foreign Key)  ExaminerId: Integer (Foreign Key)  AppearanceResults: Collection of AppearanceResult records  ScoringCriterias: Collection of ScoringCriteria records |  |
| 2 | Examiner | Manage Test Rounds | TestName, Purpose, TestType, MaxScore, DurationInMinutes, AudioFileURL, TestQuestions, TestOptions | TestName, Purpose: String (Max 50)  TestType: Enum (EnglishListening, EnglishSpeaking, Practical)  MaxScore, DurationInMinutes: Integer (≥ 0)  AudioFileURL: String (Max)  TestQuestions: Collection Of TestQuestion records  TestOptions: Collection of TestOption records |  |
| 3 | Examiner | Manage Interview Rounds | EvaluationId, EvaluationType, EvaluatedAt, Comment, ActivityId, ExaminerId, InterviewResults, InterviewCriterias | EvaluationId: Integer (Primary Key)  EvaluationType: Enum (Appearance, Interview)  Comment: String (Max 250)  IsPassed: bool  ActivityId: Integer (Foreign Key)  ExaminerId: Integer (Foreign Key)  InterviewResults: Collection of InterviewResult records  InterviewCriterias: Collection of InterviewCriteria records |  |
| 4 | Examiner | Mange Enquiry Requests | TestSessionId, Reason, Status,  TestAnswers, NewReason | TestSessonId: Integer (Foreign Key)  Reason, NewReason: String (Max 250)  Status: bool  TestAnswers: Collection of TestAnswer record |  |
| 5 | Examiner | Scoring English Speaking Test | TestSessionId,  TestAnswers, Pronunciation, Fluency, Grammar,  IsCorrect | TestSessionId: Integer (Primary Key)  TestAnswers: Collection of TestAnswer records  Pronunciation, Fluency, Grammar: Integer (≥ 0)  IsCorrect: bool |  |

#### 3.2.7 Director functional requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Actor** | **Function Name** | **Data Requirements** | **Data Validation** | **Business Rules** |
| 1 | Director | Approve/  Reject Campaign Request | RequestId, CampaignName, Desciption, JobDecription, JobRequirement, TargerQuantity  Status, RequestType, DueDate, PartnerId, DirectorId, Campaigns | RequestId: Integer (Primary Key)  CampaignName, Description: String (Max 50)  JobDescription, JobRequirement: String (Max 250)  TargetQuantity: Integer (≥ 0)  Status: Enum (Pending, Approved, Rejected, Cancelled)  RequestType: Enum (Recruitment, Promotion)  DueDate: DateOnly  PartnerId: Integer (Foreign Key)  DirectorId: Integer (Foreign Key)  Campaigns: Collection of Campaign records | BR-20 |
| 2 | Director | Approve/  Reject Campaign | CampaignId, CampaignName, CampaignType, Description, JobDescription, JobRequirement, TargetQuantity, ActualQuantity,  CampaignStatus,  StartDate, EndDate, ReviewedBy,  SeniorId, RequestId, CampaignRounds | CampaignId: Integer (Primary Key)  CampaignName, Description: String (Max 50)  CampaignType: Enum(Recruitment, Promotion)  JobDescriptoin, JobRequirement: String (Max)  TargerQuantity, ActualQuantity: Integer (≥ 0)  CampaignStatus: Enum(Draft, Pending, Approved, Rejected, Cancelled, Ongoing, Upcoming, Ended)  StartDate, EndDate: DateTime  ReviewedBy: String (Max 20)  SeniorId: Interger (Foreign Key)  RequestId: Integer (Foreign Key)  CampaignRounds: Collection of Campaign Round records |  |

#### 3.2.8 Senior Recruiter Crew functional requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Actor** | **Function Name** | **Data Requirements** | **Data Validation** | **Business Rules** |
| 1 | Senior Recruiter | Manage Campaigns | CampaignId, CampaignName, CampaignType, Description, JobDescription, JobRequirement, TargetQuantity, ActualQuantity,  CampaignStatus,  StartDate, EndDate, ReviewedBy,  SeniorId, RequestId, CampaignRounds | CampaignId: Integer (Primary Key)  CampaignName, Description: String (Max 50)  CampaignType: Enum(Recruitment, Promotion)  JobDescriptoin, JobRequirement: String (Max)  TargerQuantity, ActualQuantity: Integer (≥ 0)  CampaignStatus: Enum(Draft, Pending, Approved, Rejected, Cancelled, Ongoing, Upcoming, Ended)  StartDate, EndDate: DateTime  ReviewedBy: String (Max 20)  SeniorId: Interger (Foreign Key)  RequestId: Integer (Foreign Key)  CampaignRounds: Collection of Campaign Round records | BR-20 |
| 2 | Senior Recruiter | Manage Campaign Rounds | CampaignRoundId, RoundName,  StartDate, EndDate, Description,  TargetQuantity,  ActualQuantity,  RoundStatus,  CampaignId, Rounds | RoundName, Description: String(Max 50)  StartDate, EndDate: DateTime  TargetQuantity, ActualQuantity: Interger (≥ 0)  RoundStatus: Enum(Ongoing, Upcoming, Ended)  CampaignId: Integer (Foreign Key)  Rounds: Collection of Round records |  |
| 3 | Senior Recruiter | Manage Rounds | RoundId, Notes, RoundType, StartDate, EndDate, CampaignRoundId, TestId, RoundActivities | RoundId: Integer (Primary Key)  Notes: String (Max 50)  RoundType: Enum(Screening, Appearance, FlightHoursConfirmation, EnglishListeningTest, EnglishSpeakingTest, PracticalTest, Interview, Final)  StartDate, EndDate: DateTime  CampaignRoundId: Integer (Foreign Key)  TestId: Integer (Foreign Key)  RoundActivities: Collection of Round Activity records |  |
| 4 | Senior Recruiter | Manage Taks | AssignmentId, Task, TaskDescription, Status, AssignedAt, AssignedBy, CompletedAt, Notes, Campaignd | AssignmentId: Integer(Primary Key)  Task: Enum (Screening, Interview, TestCreation, Final)  Status: Enum (Assigned, InProgress, Completed, Cancelled)  AssignedBy: String (Max 20)  Notes: String (Max 250)  CompletedAt: DateTime  CampaignId: Integer (Foreign Key) |  |

## 4. Non-Functional Requirements

### 4.1 External Interfaces

* Web platform (ReactJS, ASP.NET Core Web API).
* Mobile App (ReactNative).
* Notifications via Email & Push (SMS optional).

### 4.2 Quality Attributes

* **Usability:** User-friendly UI with responsive design (Web & Mobile).
* **Reliability:** 99,9 % uptime, load tested to handle > 500 concurrent users.
* **Performance:** Average system response time < 3 seconds.
* **Security:** SSL Encryption, RBAC, two-factor authentication.
* **Maintainability:** Modular architecture, continous integration/deployment (CI/CD).
* **Scalability:** Support expansion to multiple airline partners in the future.

## 5. Requirement Appendix

### 5.1 Business Rules

|  |  |
| --- | --- |
| **ID** | **Rule Definition** |
| **BR-01** | Candidates must register a personal account in the system before applying for any recruitment campaign. |
| **BR-02** | The **username** must contain **8–12 characters**, using only letters and numbers. Special characters or spaces are not allowed. |
| **BR-03** | The **username** must be **unique** within the system. Duplicate usernames will trigger a system warning before submission. |
| **BR-04** | The **email address** must be in valid format (e.g., user@example.com). Invalid email formats will be rejected automatically. |
| **BR-05** | Each email must be **unique** and cannot be used to register multiple accounts. |
| **BR-06** | The **phone number** must contain **10–12 digits** and consist of numeric characters only. |
| **BR-07** | The **phone number** must be unique within the system. Duplicate phone numbers are not allowed. |
| **BR-08** | The **password** must have at least **8 characters**, including **one uppercase letter, one lowercase letter, one digit, and one special symbol**. |
| **BR-09** | All required fields (username, email, phone, password) must be completed. Missing fields trigger on-screen error messages. |
| **BR-10** | Registration attempts using identical **email, phone number, or national ID** will be automatically rejected. |
| **BR-11** | Users log in using their **registered username and password**. |
| **BR-12** | If incorrect credentials are entered more than **five consecutive times**, the account will be **temporarily locked for 15 minutes** to prevent brute-force attacks. |
| **BR-13** | If login is successful, the system redirects the user to the appropriate dashboard based on their **role** (e.g., Candidate, Recruiter, Senior Recruiter, Admin). |
| **BR-14** | If login fails, an error message is displayed: “Invalid username or password.” No hint about which field is incorrect is provided for security reasons. |
| **BR-15** | Users who forget their password can use the **“Forgot Password”** feature to reset via verified email |
| **BR-16** | Candidates must **complete and submit their online application** for the selected campaign **before the submission deadline**. Only fully submitted applications will be considered for screening. |
| **BR-17** | Applications will only be approved if all required documents (ID/Passport, certificates, portrait photo, CV) are submitted and meet the Academy’s standards. |
| **BR-18** | Candidates may edit their application before the deadline. Once the application is approved, the information will be locked to ensure accuracy. |
| **BR-19** | Only candidates with approved applications will receive interview schedules or be allowed to take online tests. |
| **BR-20** | Application forms must comply with standardized templates provided by the Academy. |
| **BR-21** | Late applications submitted after the submission period ends will not be considered under any circumstances and will be assigned a “rejected” status. |
| **BR-22** | Only **PDF** and **JPEG** file formats are accepted, and each file must not exceed **5 MB** in size. |
| **BR-23** | Candidates may not submit multiple applications within the same recruitment campaign round. |
| **BR-24** | Recruitment rounds must follow a fixed order: Application Screening → Appearance Check → English Test → Interview → Final Result. |
| **BR-25** | Candidates must achieve a minimum score in rounds such as appearance assessment, English proficiency test, and interview round (e.g. ≥ 70/100) to move on to the next round. |
| **BR-26** | If a candidate is absent without prior notice during the interview round, the system will automatically cancel that candidate's schedule and mark the status as "Withdrawn". |
| **BR-27** | Recruitment exams are English proficiency tests, candidates must pass all sections to qualify for the next round |
| **BR-28** | After the final results of the rounds are available, Candidates must provide an accurate health declaration for confirmation; false information will result in disqualification. |
| **BR-29** | Each campaign must consist of **a minimum of two to a maximum of three rounds**, defined by the Senior Recruiter when creating and sending the campaign to Recruiters. |
| **BR-30** | Each batch within a campaign shall recruit approximately **60–70%** of the total target. Subsequent batches will allocate **60–70% of the remaining quota** until the campaign reaches completion. |
| **BR-31** | When the campaign end date is reached, if the number of recruited candidates is lower than the total quota, the campaign status will automatically change to **“Ended – Quota Not Met.”** The system will display both **Total Target** and **Actual Recruited** values for review. |
| **BR-32** | If all batches are completed and the campaign still fails to meet the total target, the system shall alert the **Airline Partner** and **Senior Recruiter** to decide whether to **extend**, **transfer remaining quota**, or **close** the campaign as **“Completed with Deficit.”** |
| **BR-33** | Candidates to **apply for only one active campaign** at a time |
| **BR-34** | Cabin crew members must complete a minimum service period (e.g., 12 months) before being eligible for promotion exams. |
| **BR-35** | Promotion exams are tests of aviation expertise, how to handle real-life situations, flight attendants must pass all sections to qualify for the next round. |
| **BR-36** | Promotion results are determined by aggregating the results of the rounds and confirming the health check report. |
| **BR-37** | Promotion eligibility must be revalidated annually to reflect updated Academy policies. |
| **BR-38** | Promotion requests will be denied if the employee has unresolved disciplinary records. |
| **BR-39** | Scores are entered directly into the system by the assigned examiners and cannot be modified once confirmed. |
| **BR-40** | Final results are automatically calculated based on predefined weightings and published simultaneously for all candidates. |
| **BR-41** | The system automatically sends notifications (via email or app) whenever there are important updates, such as approval/rejection, exam schedules, or result announcements. |
| **BR-42** | The system must store the entire recruitment and promotion history, including “upcoming”, “in progress”, “ended” status, for reporting purposes and campaign aggregation. |
| **BR-43** | Only Senior Recruiters or Recruiters are allowed to create, edit, or close recruitment campaigns. |
| **BR-44** | Only Senior Recruiters or Recruiters are allowed to create, edit, or close promotion campaigns. |
| **BR-45** | Only Senior Recruiters may assign examiners, schedule interviews, and activate exam sessions. |
| **BR-46** | Only Examiners   assigned to a particular round can enter scores for that round. |
| **BR-47** | The system shall automatically notify candidates if required documents are missing. |
| **BR-48** | Any discrepancy between recorded data and manual records must trigger a system alert and audit log entry. |
| **BR-49** | Interview panels must consist of three or five  examiners to ensure impartiality. |
| **BR-50** | Evaluation criteria must remain consistent across all candidates within the same campaign. |
| **BR-51** | Subjective evaluation comments must accompany numerical scores for transparency. |
| **BR-52** | Examiners must finalize and submit their assessments within 24 hours of the interview. |
| **BR-53** | Candidates failing any mandatory criterion cannot proceed, regardless of overall score. |
| **BR-54** | All recruitment campaigns must follow the predefined workflow without alteration. |
| **BR-55** | Disqualified applicants must receive a written explanation via email. |
| **BR-56** | Once started, online exams must be completed in one sitting; no pauses are permitted. |
| **BR-57** | The system must automatically submit answers once the allocated time expires. |
| **BR-58** | Exam retakes are not permitted unless explicitly authorized by Senior Recruiters/Recruiters due to technical issues. |
| **BR-59** | Randomized question banks must be used to minimize cheating. |
| **BR-60** | Results of online exams must be available within 48 hours of completion. |
| **BR-61** | Examiners cannot access candidate responses until the exam session closes. |
| **BR-62** | Cheating or rule violations will result in immediate disqualification. |
| **BR-63** | Exam system logs must be archived for a minimum of three years. |
| **BR-64** | English tests can be used for one or more recruitment rounds, this round can use this test and the next round can use another test |
| **BR-65** | Proficiency tests can have random question format to avoid doing the same test again |
| **BR-66** | All candidates must undergo a **physical measurement check** as part of the initial evaluation round. |
| **BR-67** | Each measurement record must include Height **(cm)**, **Weight (kg)**, and **BMI**, calculated automatically by the system based on entered data. |
| **BR-68** | The system must validate that all physical measurement fields are completed and within acceptable ranges before submission. |
| **BR-69** | Candidates’ **BMI** must fall within the range of **18.5 – 24.9** to be considered “Healthy.” Any out-of-range results must be flagged for manual review. |
| **BR-70** | Minimum physical requirements differ by gender and will be automatically verified during data entry:  • **Male**: Height ≥ 170cm, Weight ≥ 55kg  • **Female**: Height ≥ 160cm, Weight ≥ 45kg |
| **BR-71** | Candidates must have **no visible tattoos, scars, or deformities** affecting appearance |
| **BR-72** | All physical measurement records must be confirmed by the authorized recruiter or examiner within **24 hours** after measurement. |
| **BR-73** | The system must automatically mark physical assessment status as “Incomplete” if any measurement or validation step is missing. |
| **BR-74** | Candidates failing the physical measurement requirements are automatically **disqualified** from proceeding to later rounds (English Test, Interview, etc.). |
| **BR-75** | During the promotion evaluation process, unlike recruitment, flight attendants must be verified and their flight hours compared, only those who meet the latest standards will continue, without the need for a physical re-evaluation. |
| **BR-76** | Candidates’ physical results will be combined with **CatwalkScore** and **Appearance Evaluation** to form the overall “Appearance Grade.” |
| **BR-77** | No authorized person may modify or delete physical measurement entries after submission. |
| **BR-78** | For audit purposes, physical measurement data must be stored for at least **5 years**. |
| **BR-79** | If automatic validation detects **out-of-range or missing measurement data** (e.g., height, weight, BMI), the system will prompt the examiner to **review and confirm** the values before submission.  Only numeric values within the valid range defined by the campaign standards will be accepted. |
| **BR-80** | Airline Partners can submit new campaign requests specifying campaign type, target quantity, and description. |
| **BR-81** | Senior Recruiter must review and validate campaign details before forwarding them to the Director. |
| **BR-82** | The Director must approve or reject each campaign request through the system interface by selecting the corresponding action button. The decision will be automatically recorded and reflected in the campaign’s status. |
| **BR-83** | Only approved campaigns can be activated and assigned to Recruiters. |
| **BR-84** | Rejected campaigns may be resubmitted after revision and must go through the full approval process again. |
| **BR-85** | The system notifies relevant users (Airline Partner, Senior Recruiter, Recruiter) **only when the final campaign result is confirmed**, such as campaign completion or closure |
| **BR-86** | Promotion requests must be reviewed and approved by the Director or Senior Recruiter before the system schedules the exam. |

### 5.2 Common Requirements

* The system must support both desktop and mobile access.
* Notifications must be sent automatically for every status update.
* All forms must validate required fields before submission.

### 5.3 Application Messages List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Message Code** | **Message Type** | **Context** | **Content** |
| 1 | MSG01 | In line | There is not any search result | *No search results* |
| 2 | MSG02 | Toast message | Updating profile’s information successfully | *Update successfully* |
| 3 | MSG01 | In line | There is not any search result | *No search results* |
| 4 | MSG01 | In line | There is not any search result | *No search results* |
| 5 | MSG01 | In line | There is not any search result | *No search results* |
| 6 | MSG01 | In line | There is not any search result | *No search results* |
| 7 | MSG01 | In line | There is not any search result | *No search results* |
| 8 | MSG01 | In line | There is not any search result | *No search results* |
| 9 | MSG01 | In line | There is not any search result | *No search results* |

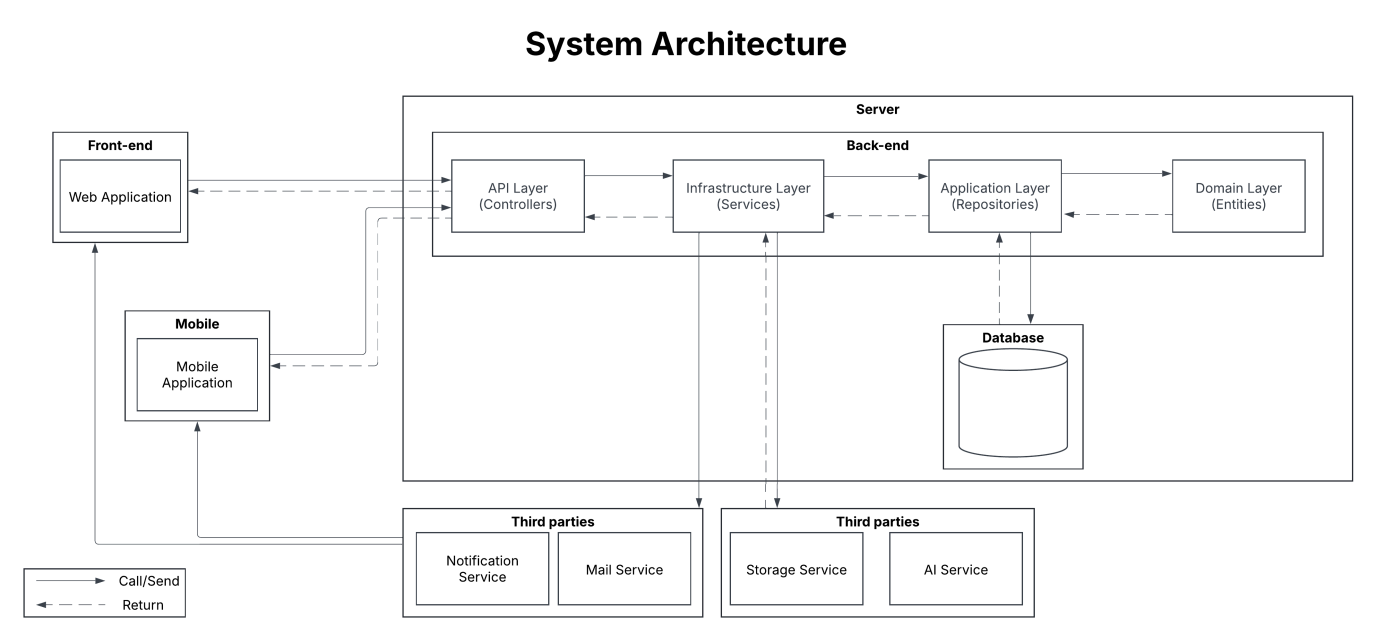
### 5.4 Other Requirements…

* The system must support bilingual content (English & Vietnamese).
* Logs must be exportable for auditing purposes.
* Mobile apps must support both iOS and Android.

# IV. Software Design Description

## 1. System Design

### 1.1 System Architecture

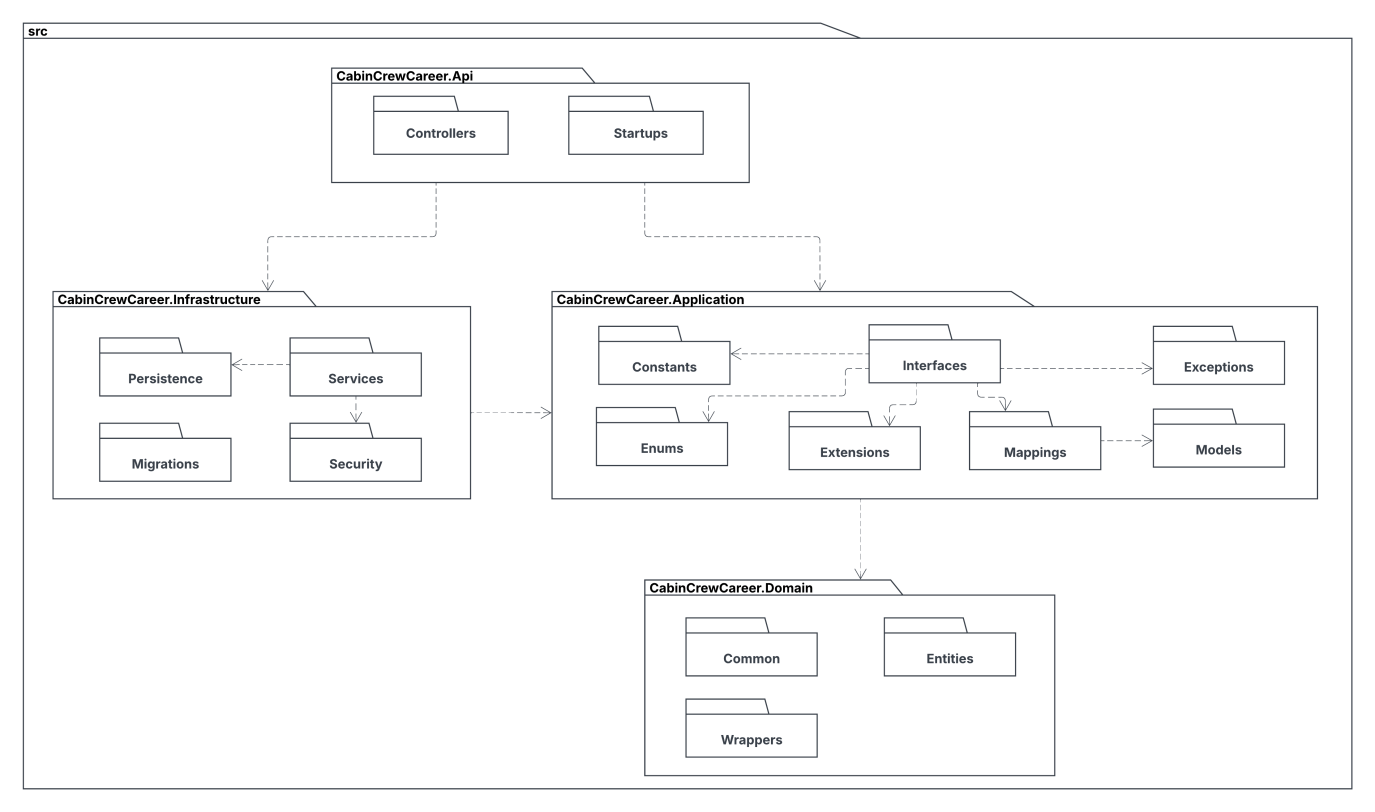


### 1.2 Package Diagram

#### 1.2.1 Front-end

***Packages Desciption***

#### 1.2.2 Back-end



***Packages Desciption***

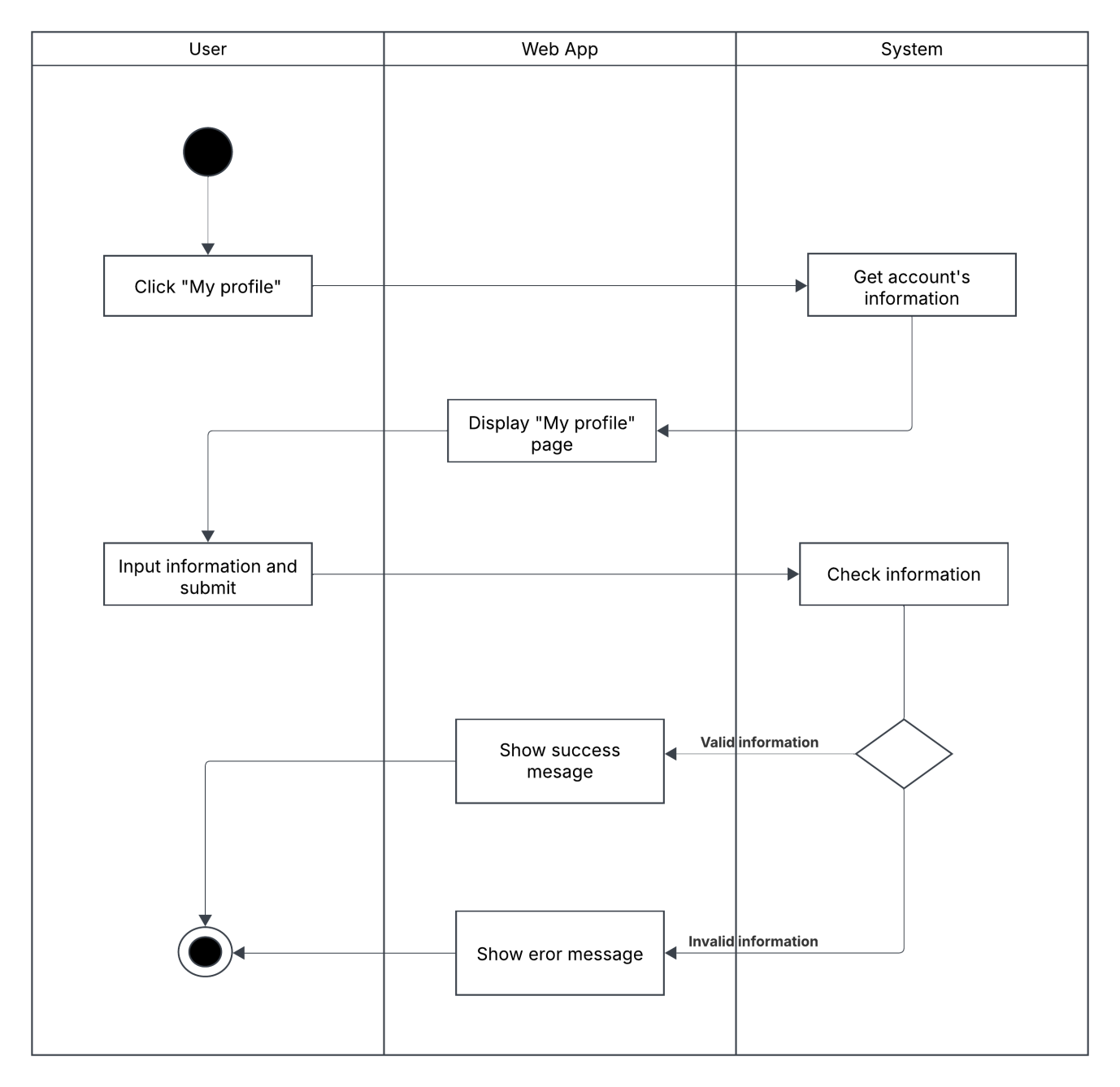
|  |  |  |
| --- | --- | --- |
| **No.** | **Package** | **Description** |
| 1 | Controllers | This package contains controller classes in app. |
| 2 | Startups | This package contains start up classes such as Swagger in app. |
| 3 | Persistences | This package contains general and database related classes. |
| 4 | Services | This packages contains common service classes. |
| 5 | Migrations | This package contains classes used for migrating database. |
| 6 | Security | This package contains classes used for security purposes. |
| 7 | Constants | This package contains classes for common constants. |
| 8 | Interfaces | This package contains all the interfaces. |
| 9 | Exceptions | This package contains classes used for error handling purposes. |
| 10 | Enums | This package contains reponse codes as enum data type. |
| 11 | Extensions | This package contains extension classes. |
| 12 | Mappings | This package contains mapper class using for mapping profile. |
| 13 | Models | This package contains classes to map objects to/from json from API. |
| 14 | Common | This package contains enum data type used for entity classes. |
| 15 | Entities | This package contains entity classes. |
| 16 | Wrappers | This package contains wrapper classes used for request/response. |

### 1.3 Activity Diagrams

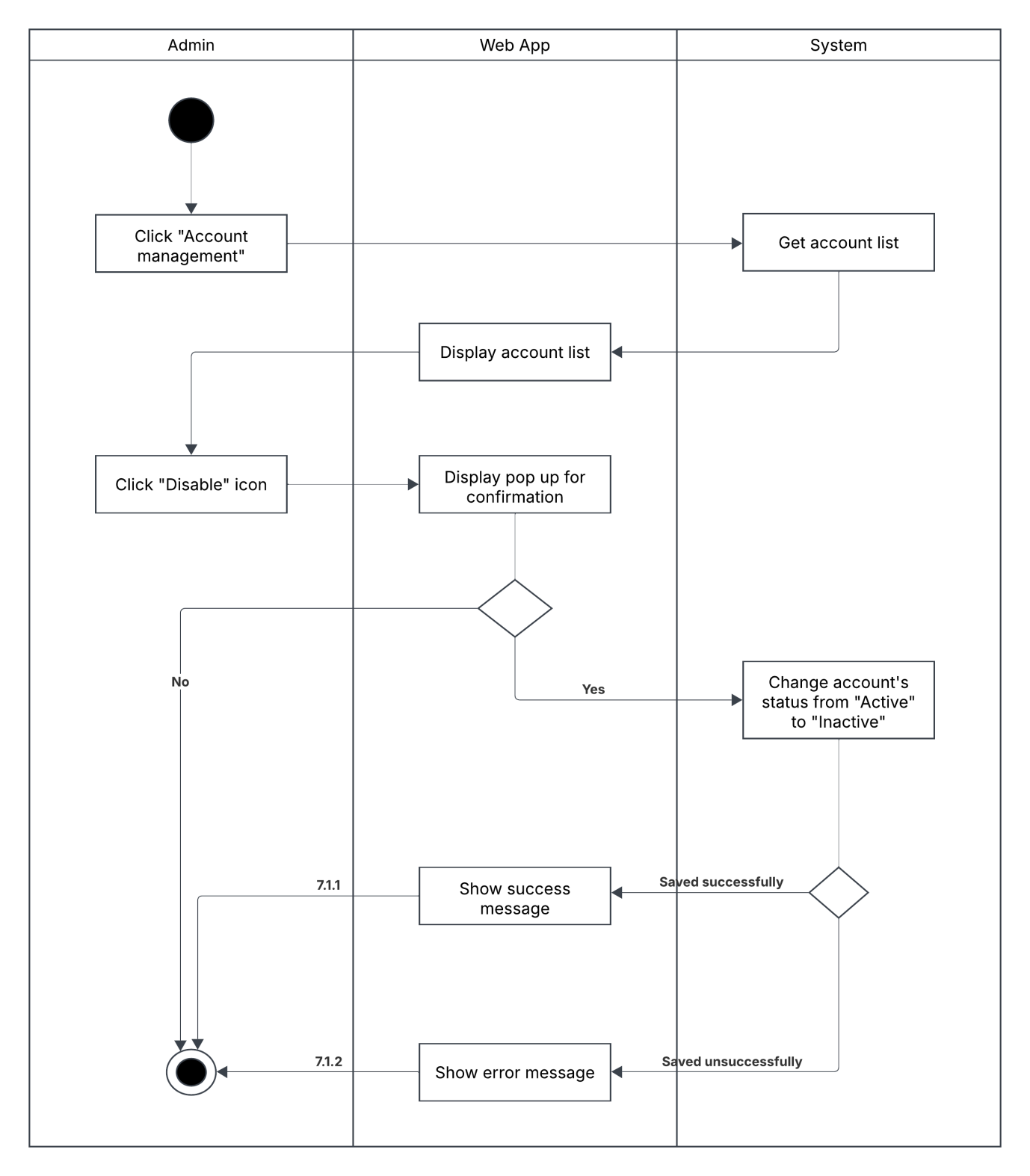
#### 1.3.1 Login Activity Diagram



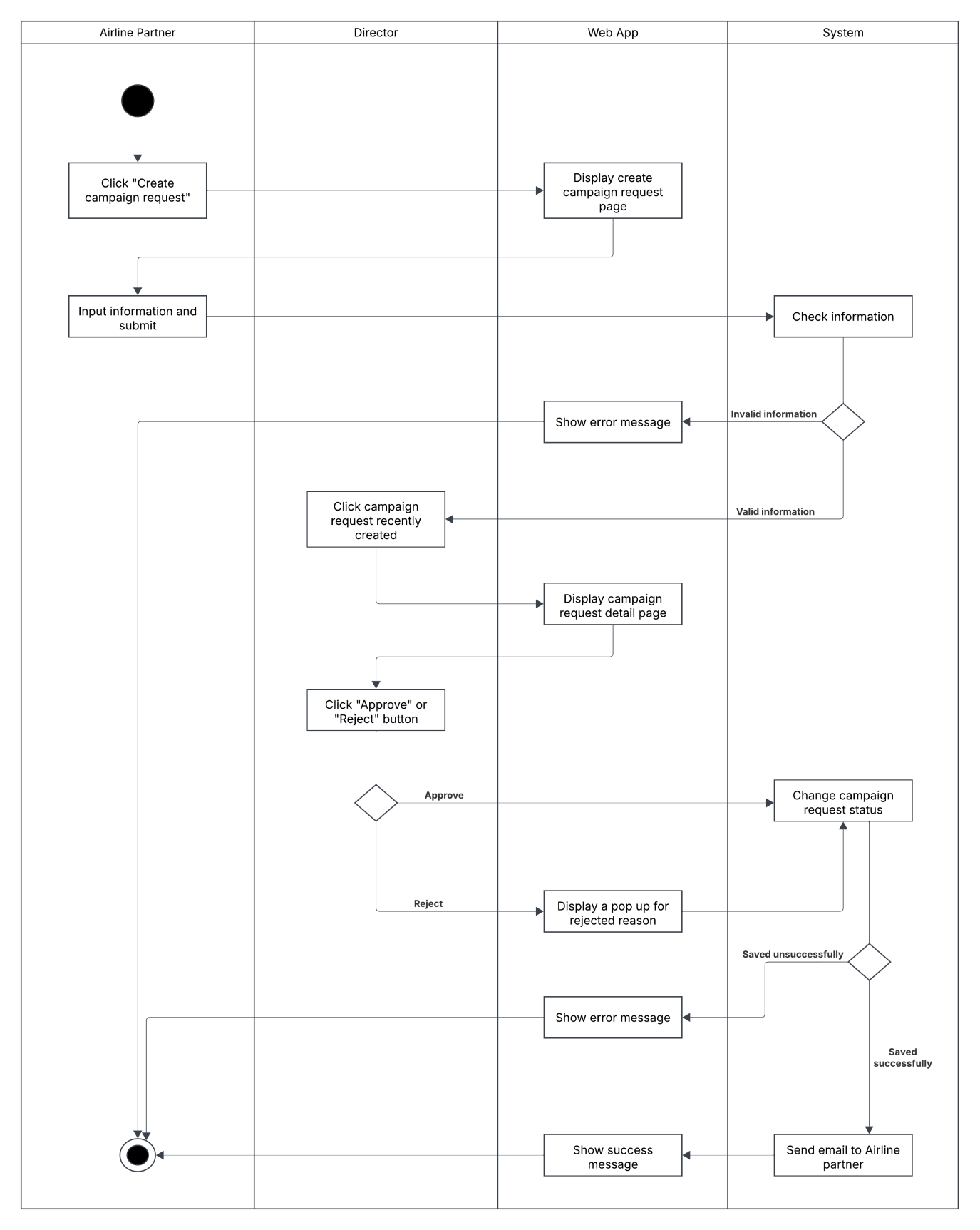
#### 1.3.2 Update Profile Activity Diagram



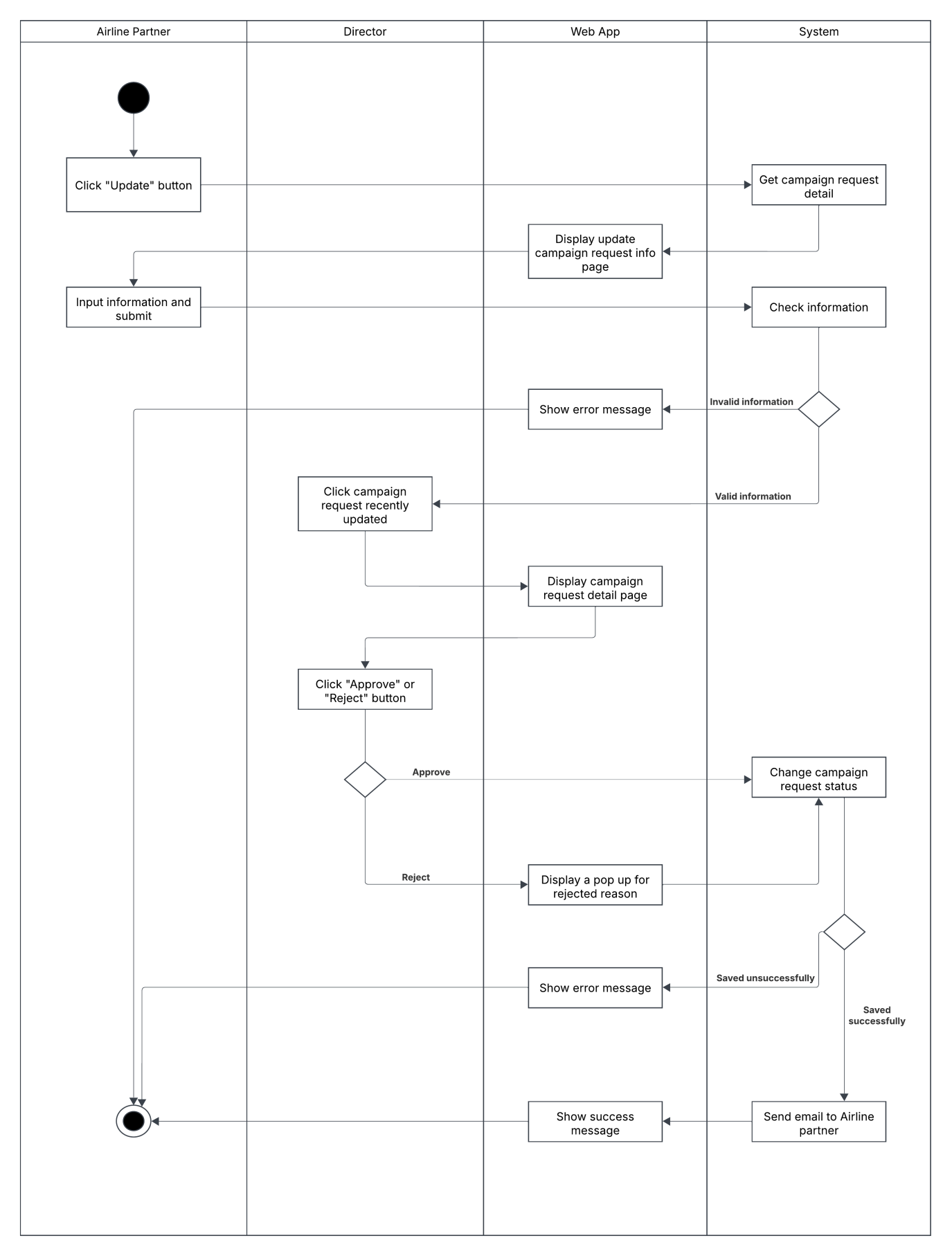
#### 1.3.3 Disable Account Activity Diagram



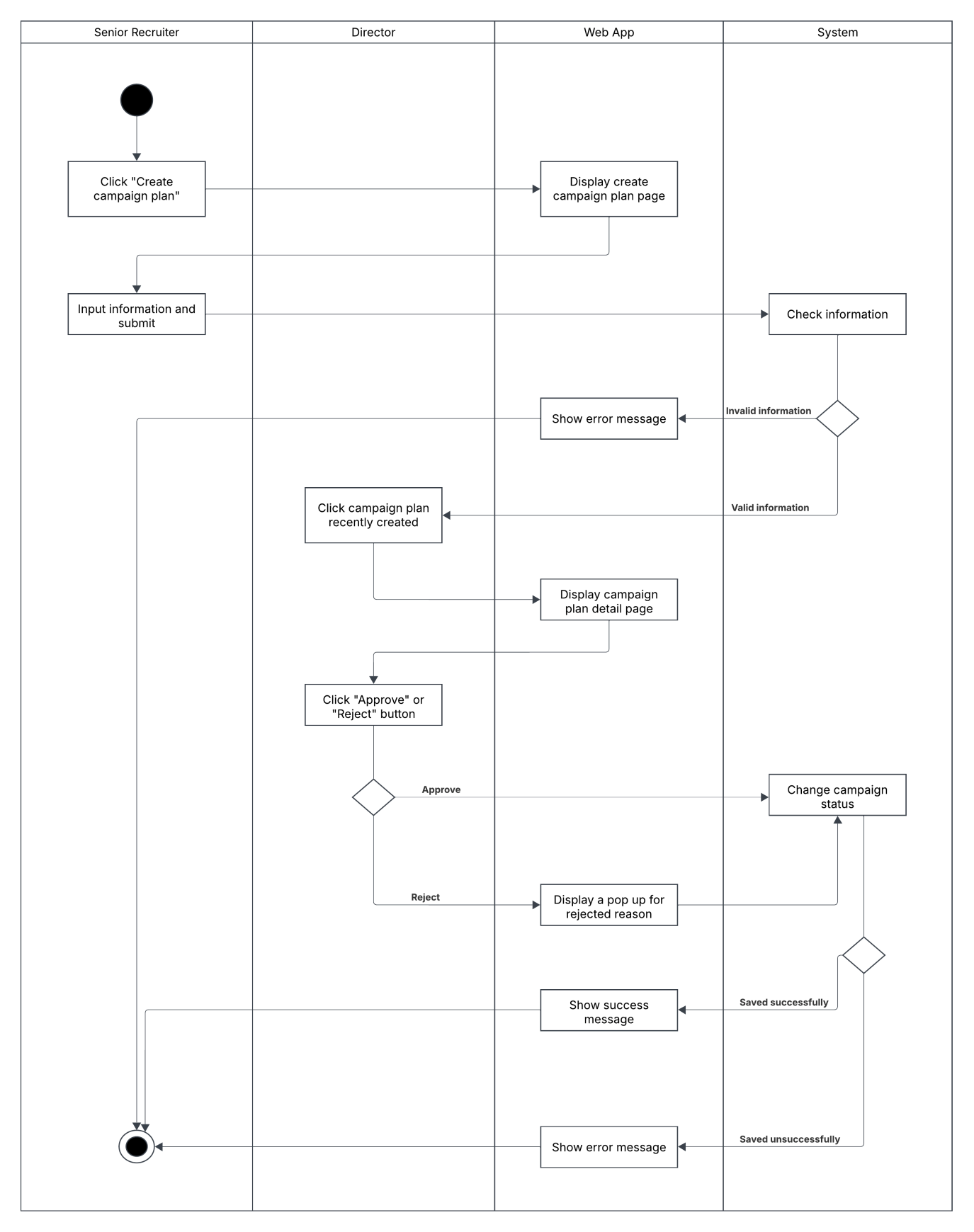
#### 1.3.4 Create Campaign Request Activity Diagram



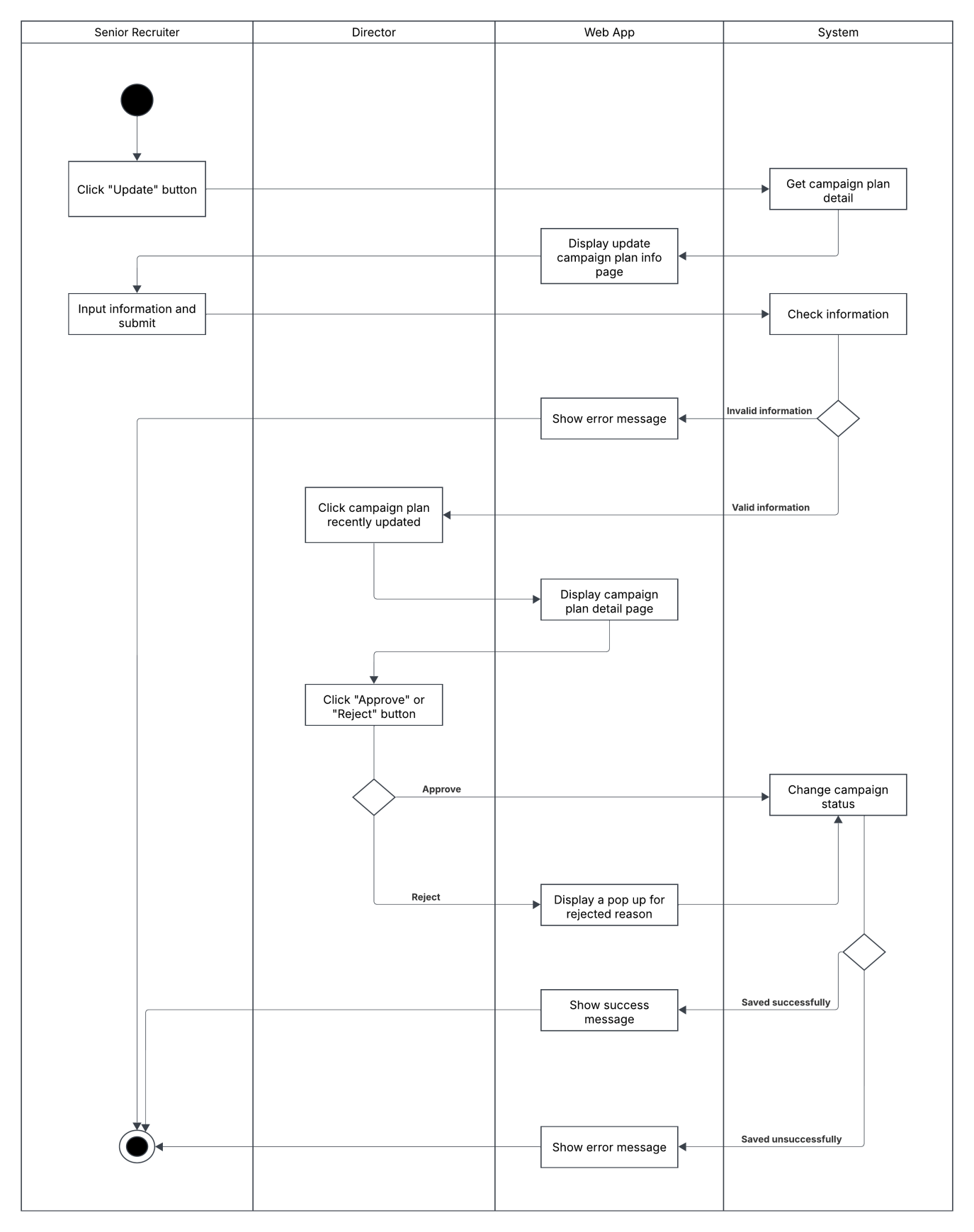
#### 1.3.5 Update Campaign Request Activity Diagram



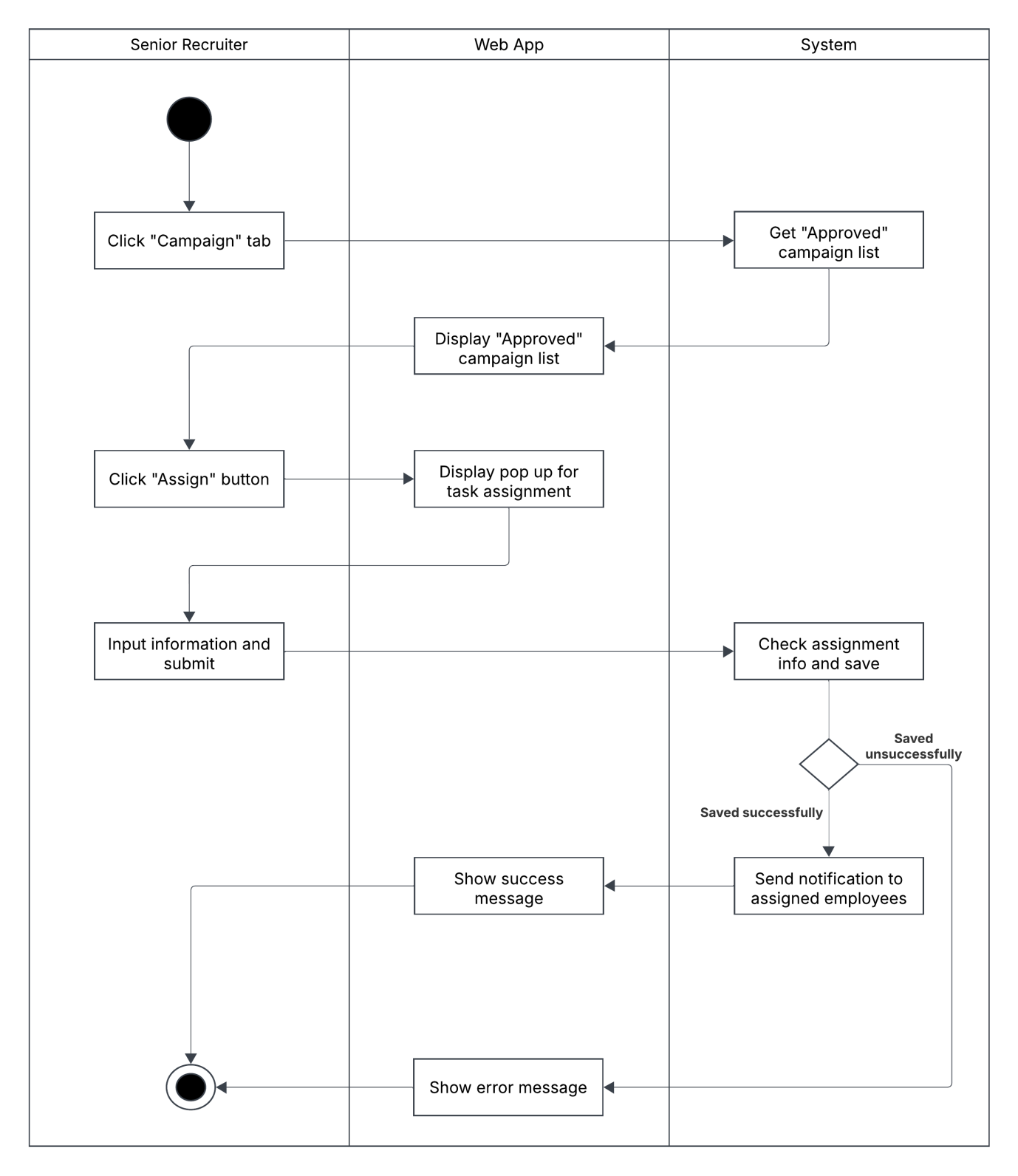
#### 1.3.6 Create Campaign Activity Diagram



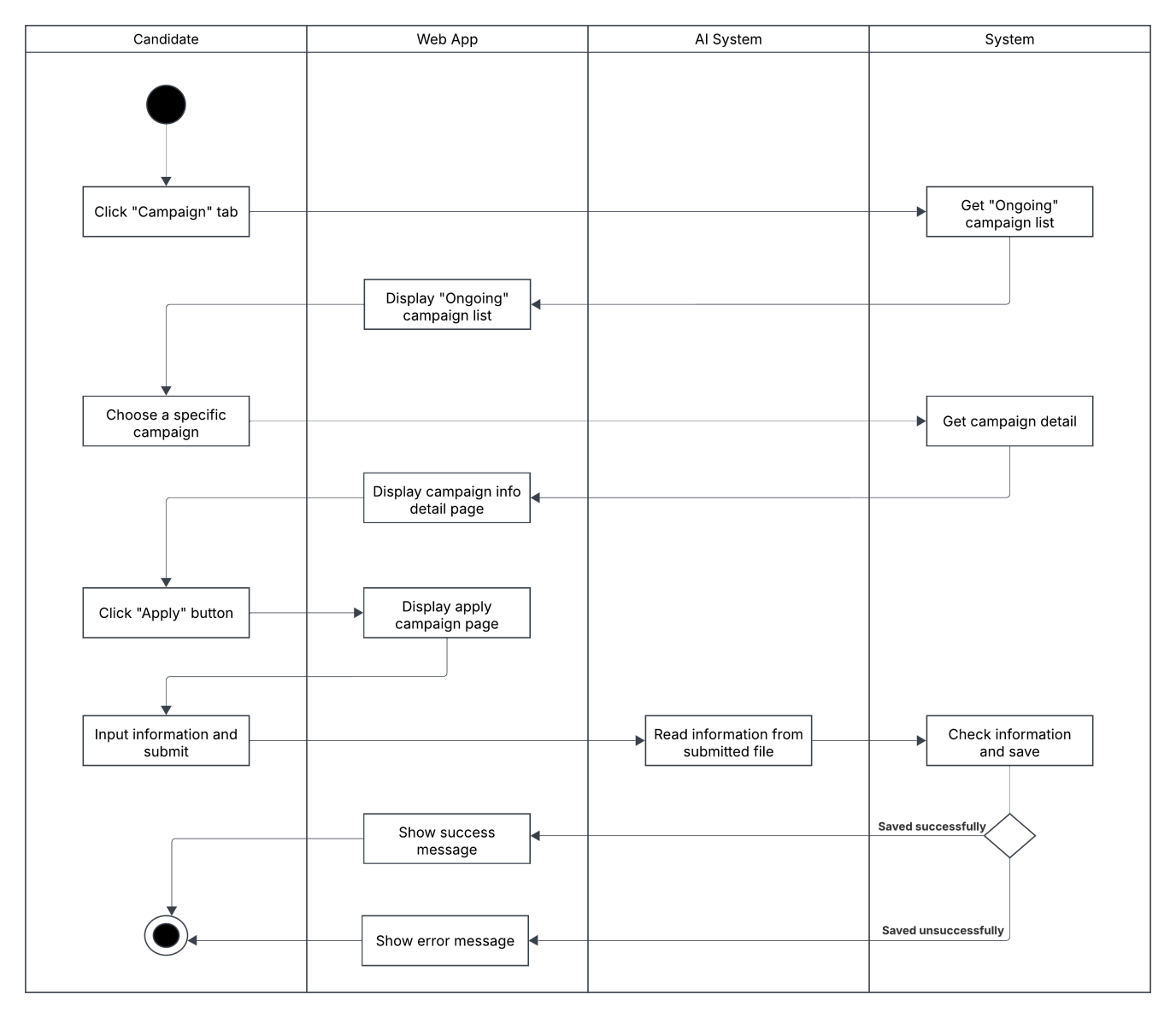
#### 1.3.7 Update Campaign Activity Diagram



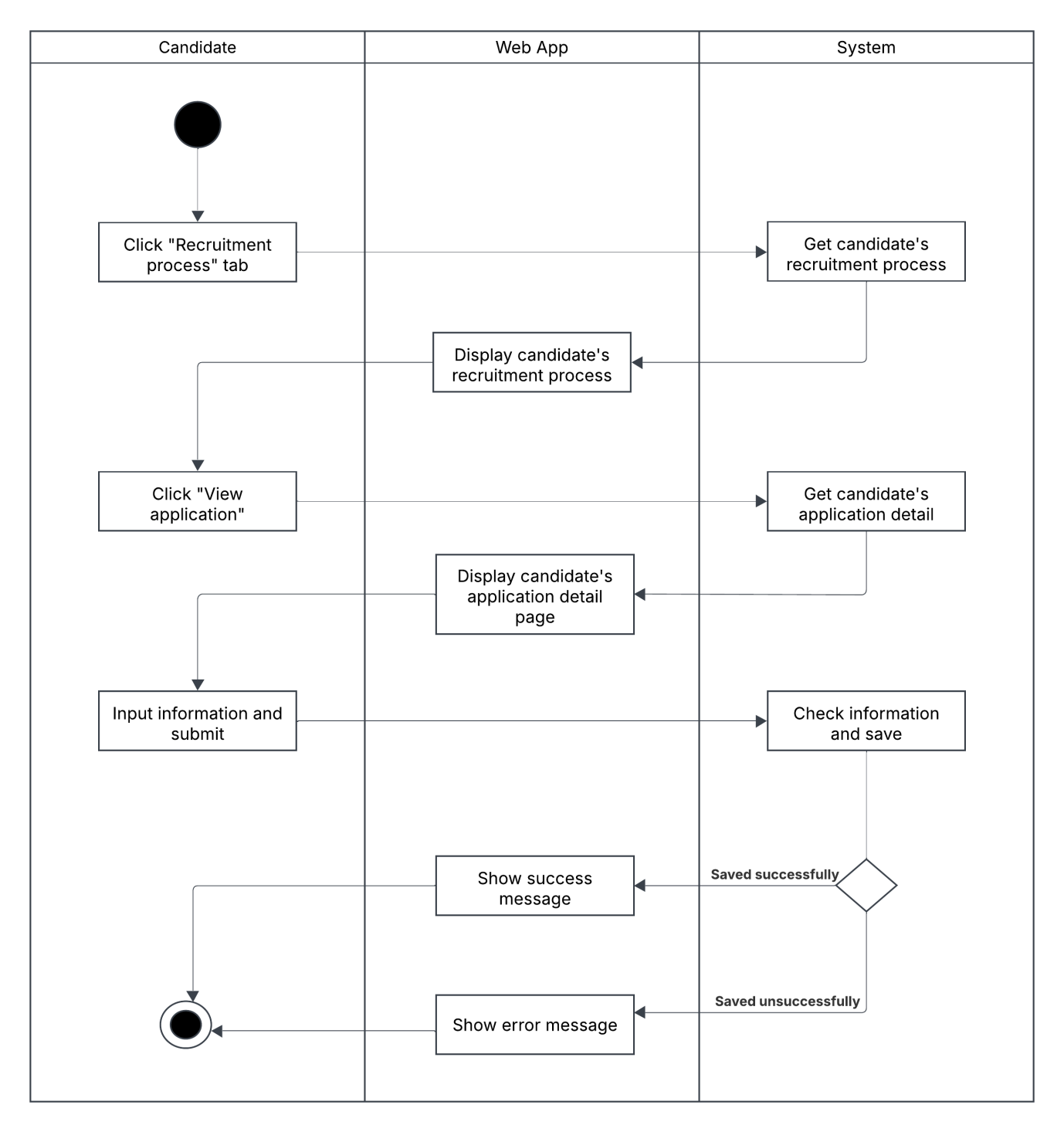
#### 1.3.8 Campaign’s Task Assignment Activity Diagram



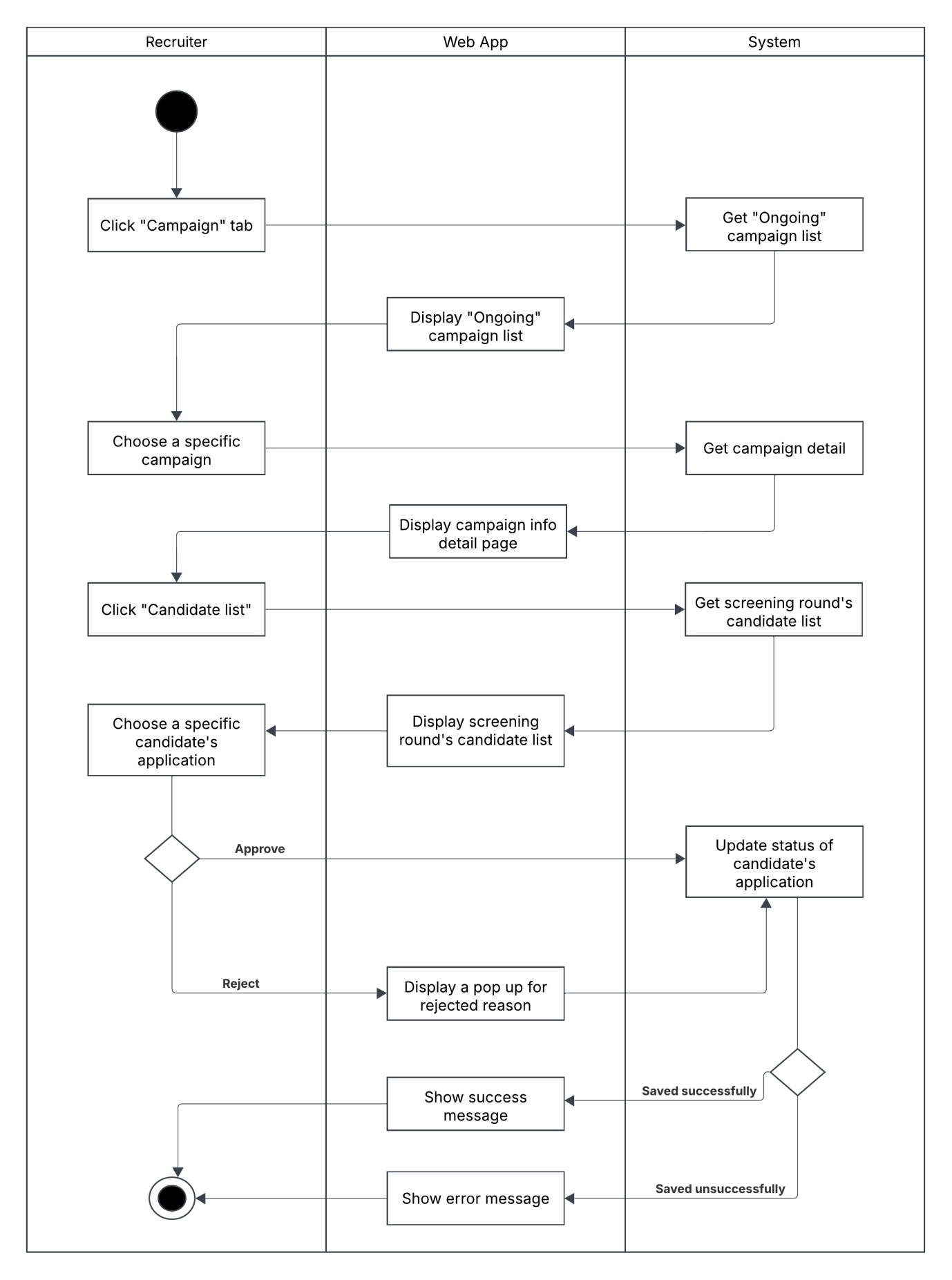
#### 1.3.9 Apply To Campaign Round Activity Diagram



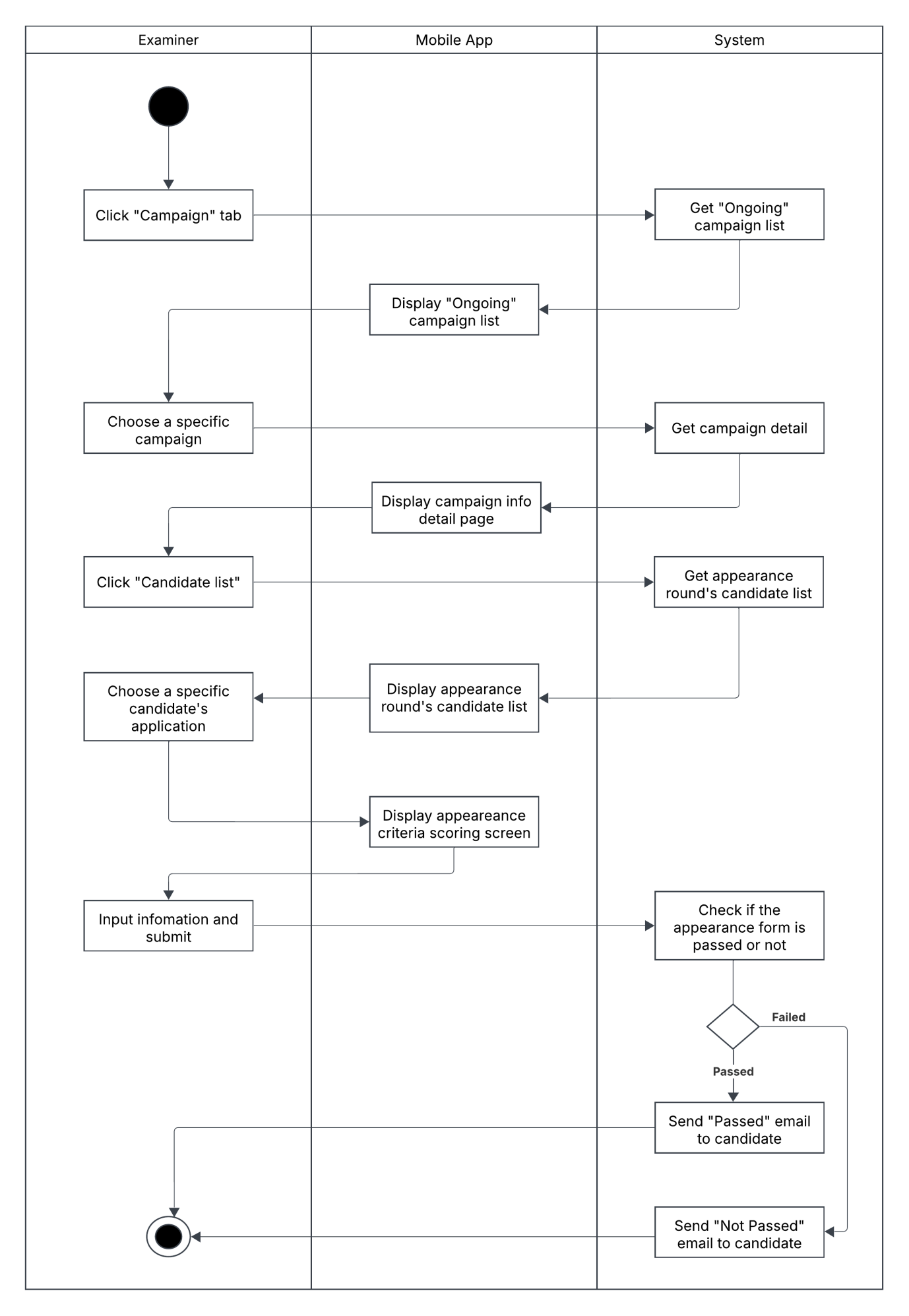
#### 1.3.10 Update Candidate’s Application Activity Diagram



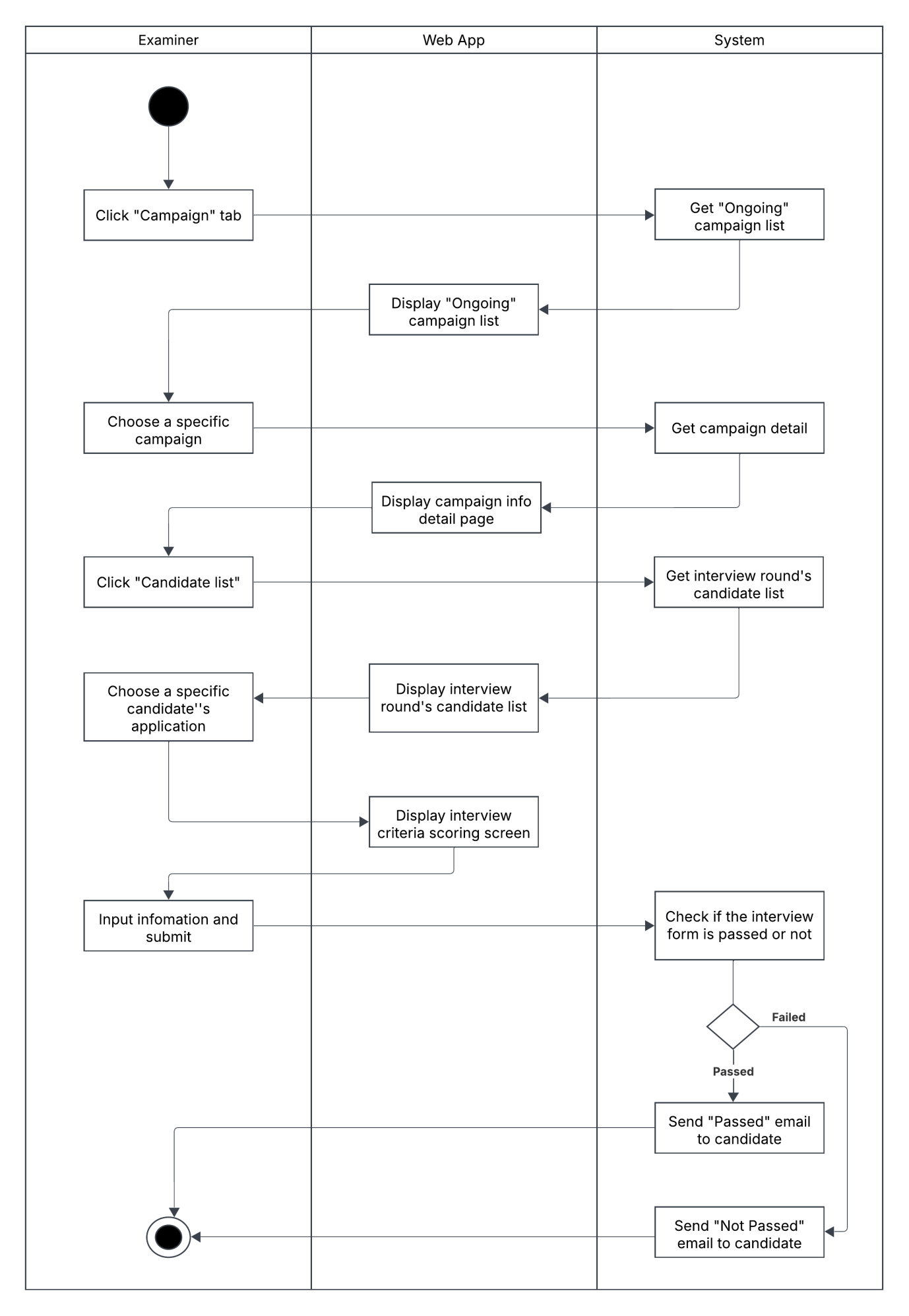
#### 1.3.11 Screening Round Evaluation Activity Diagram



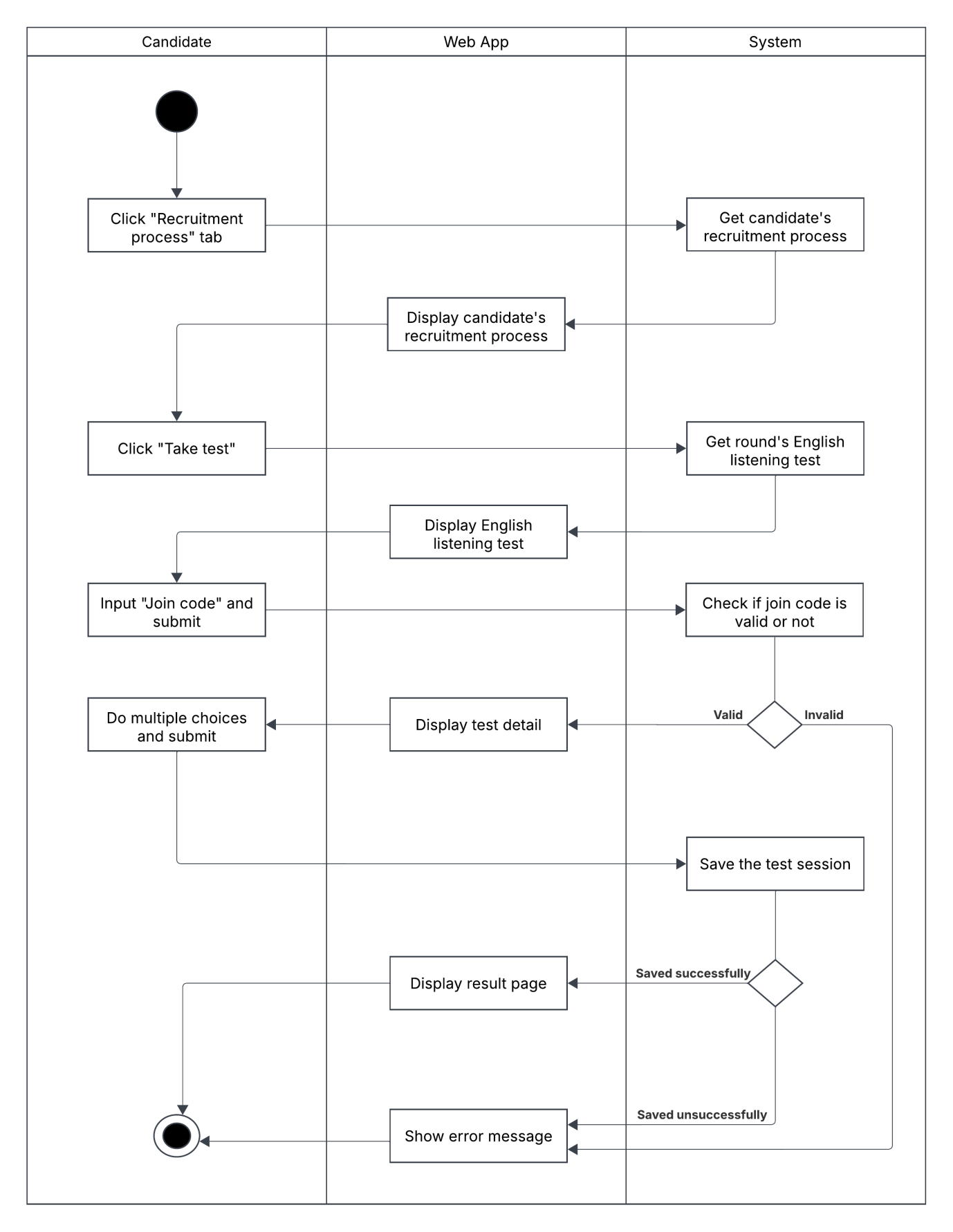
#### 1.3.12 Create Appearance Evaluation Form Activity Diagram



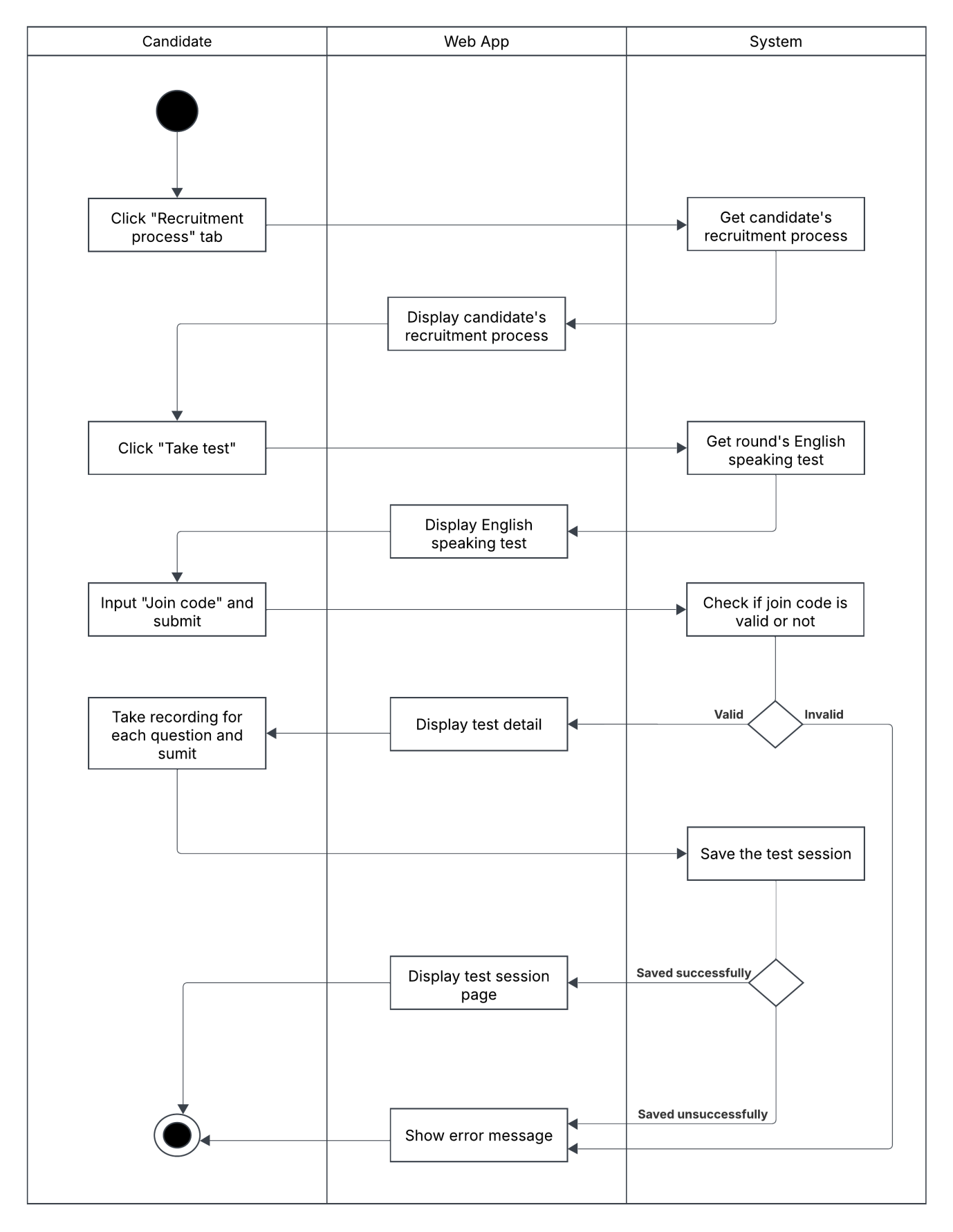
#### 1.3.13 Create Interview Evaluation Form Activity Diagram



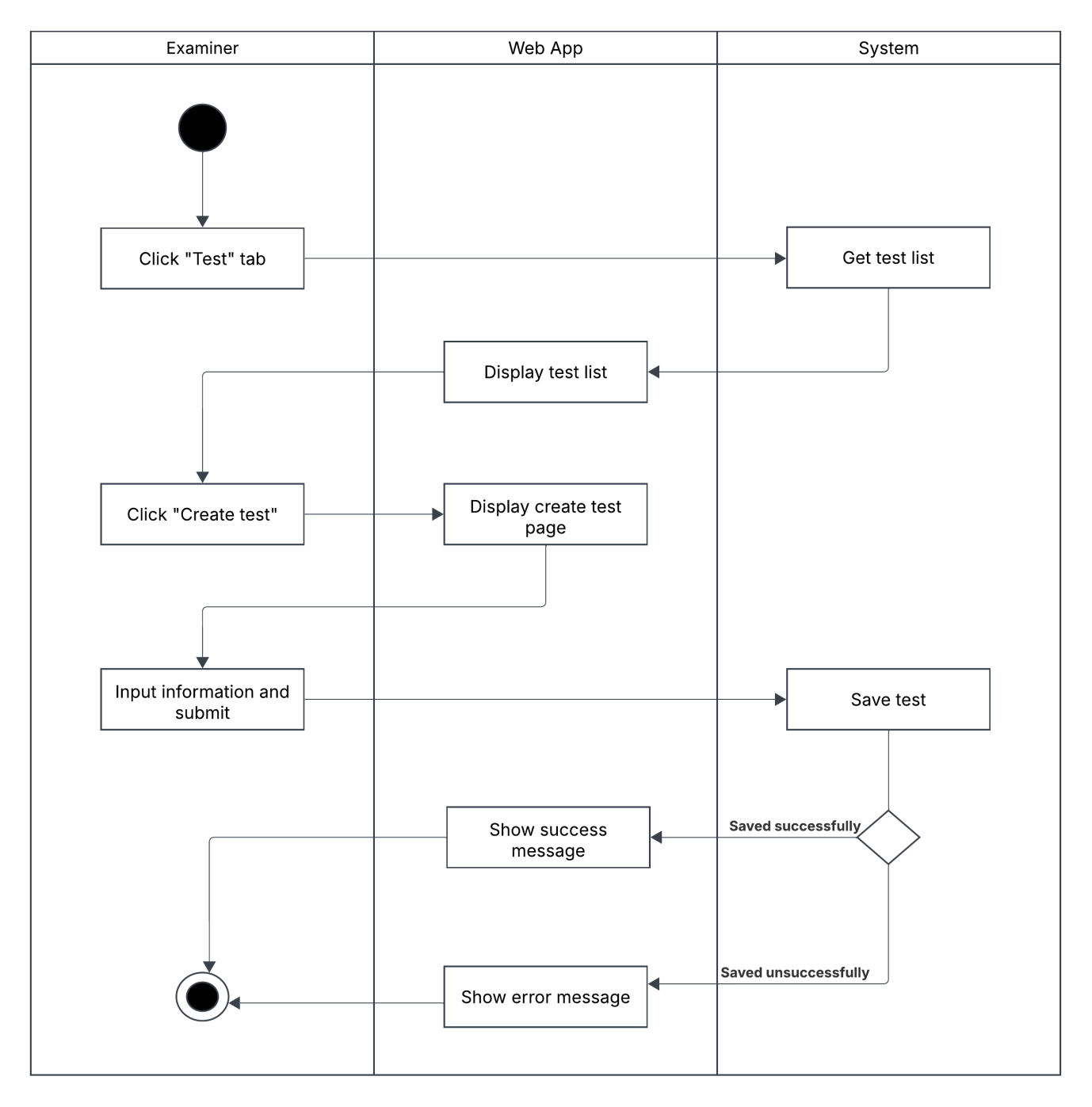
#### 1.3.14 Take English Listening Test Activity Diagram



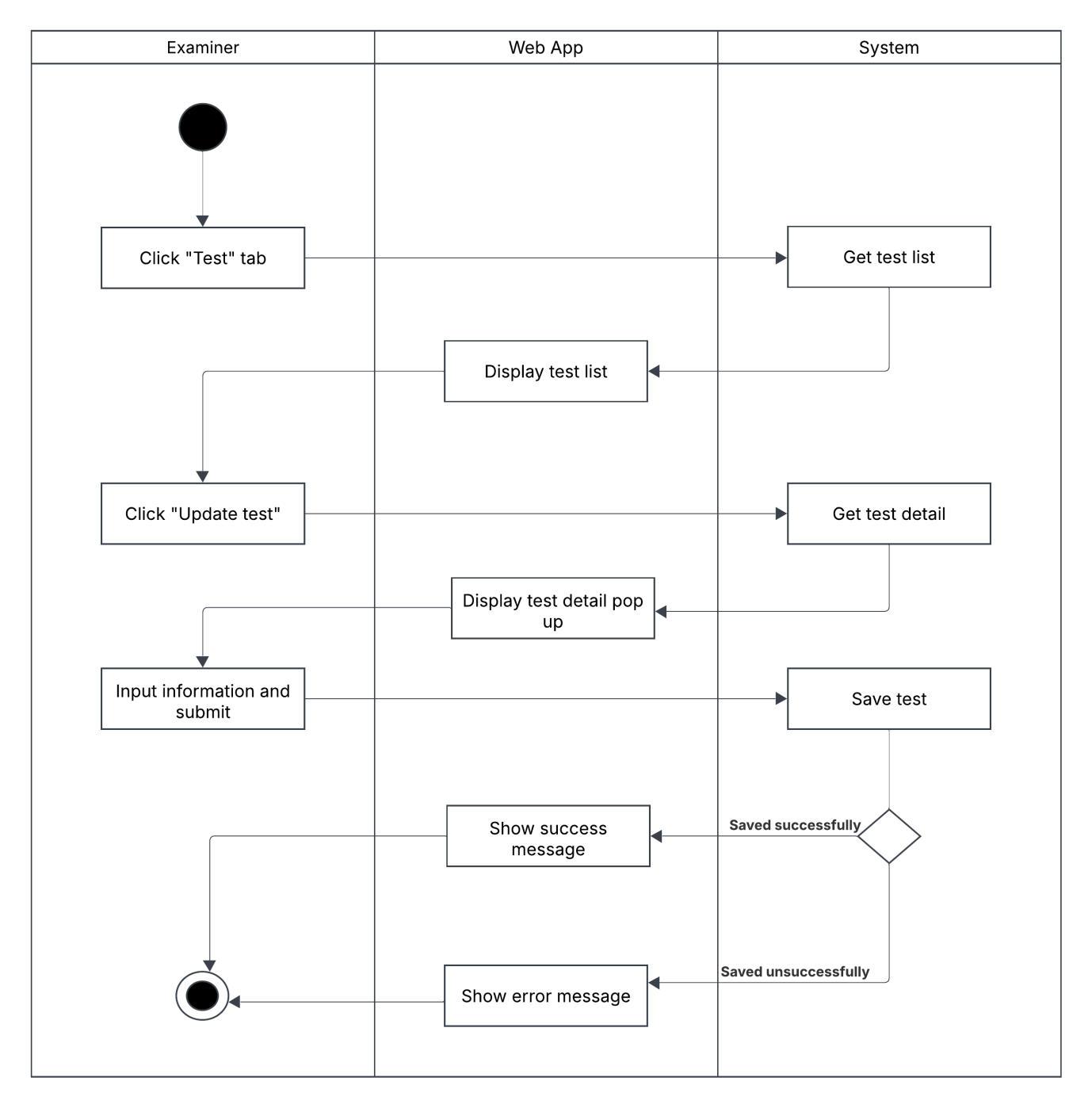
#### 1.3.15 Take English Speaking Test Activity Diagram



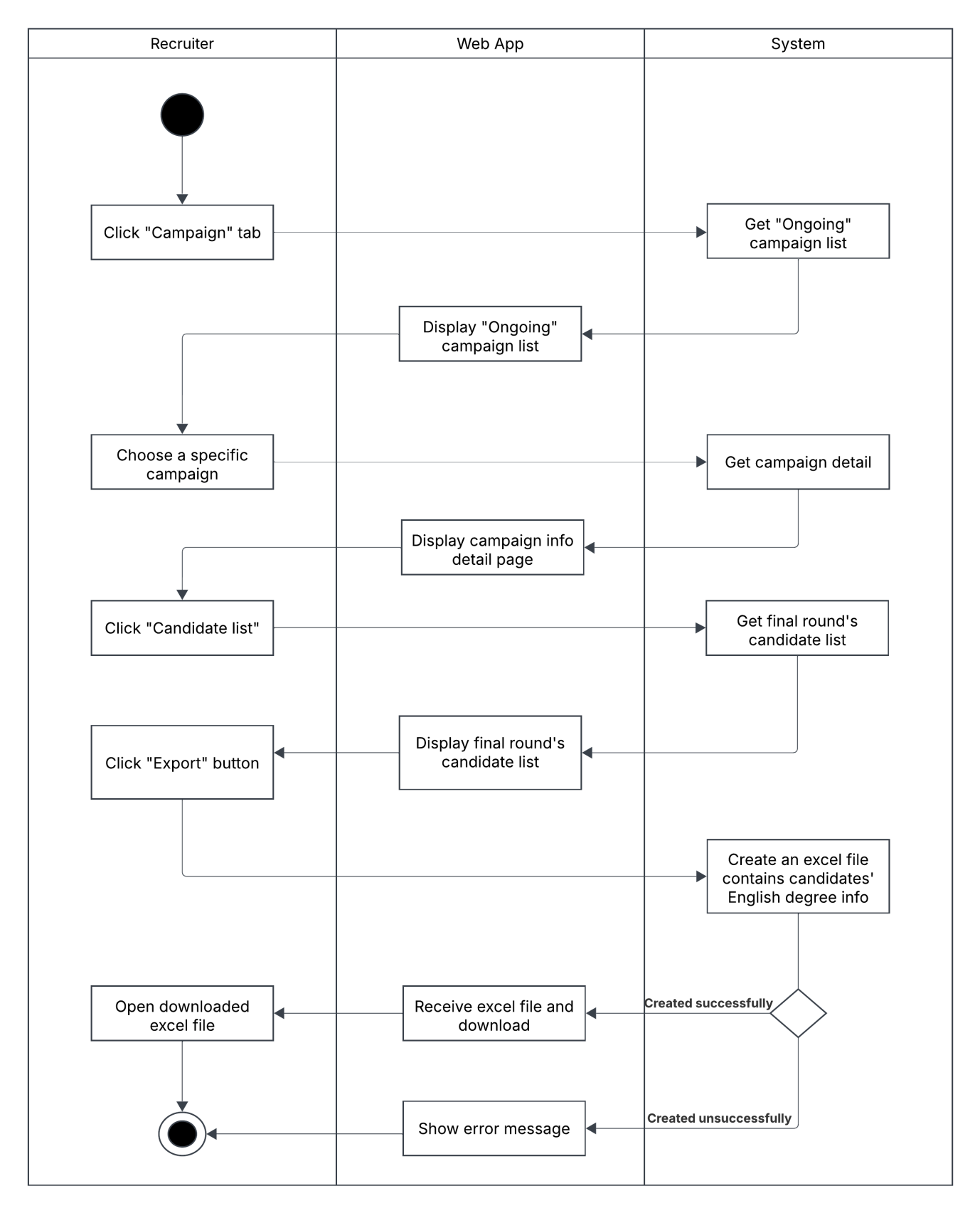
#### 1.3.16 Create Test Activity Diagram



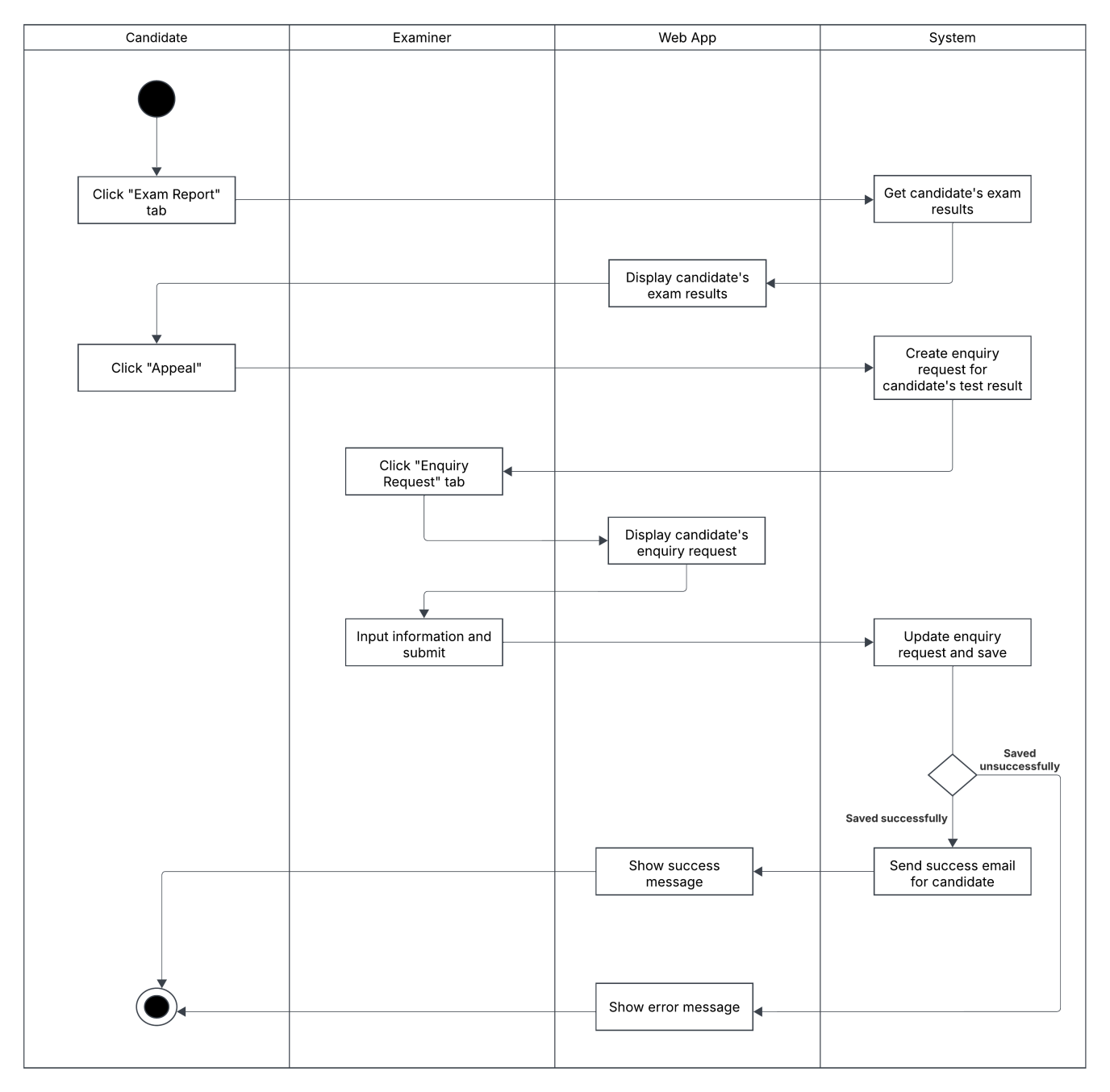
#### 1.3.17 Update Test Activity Diagram



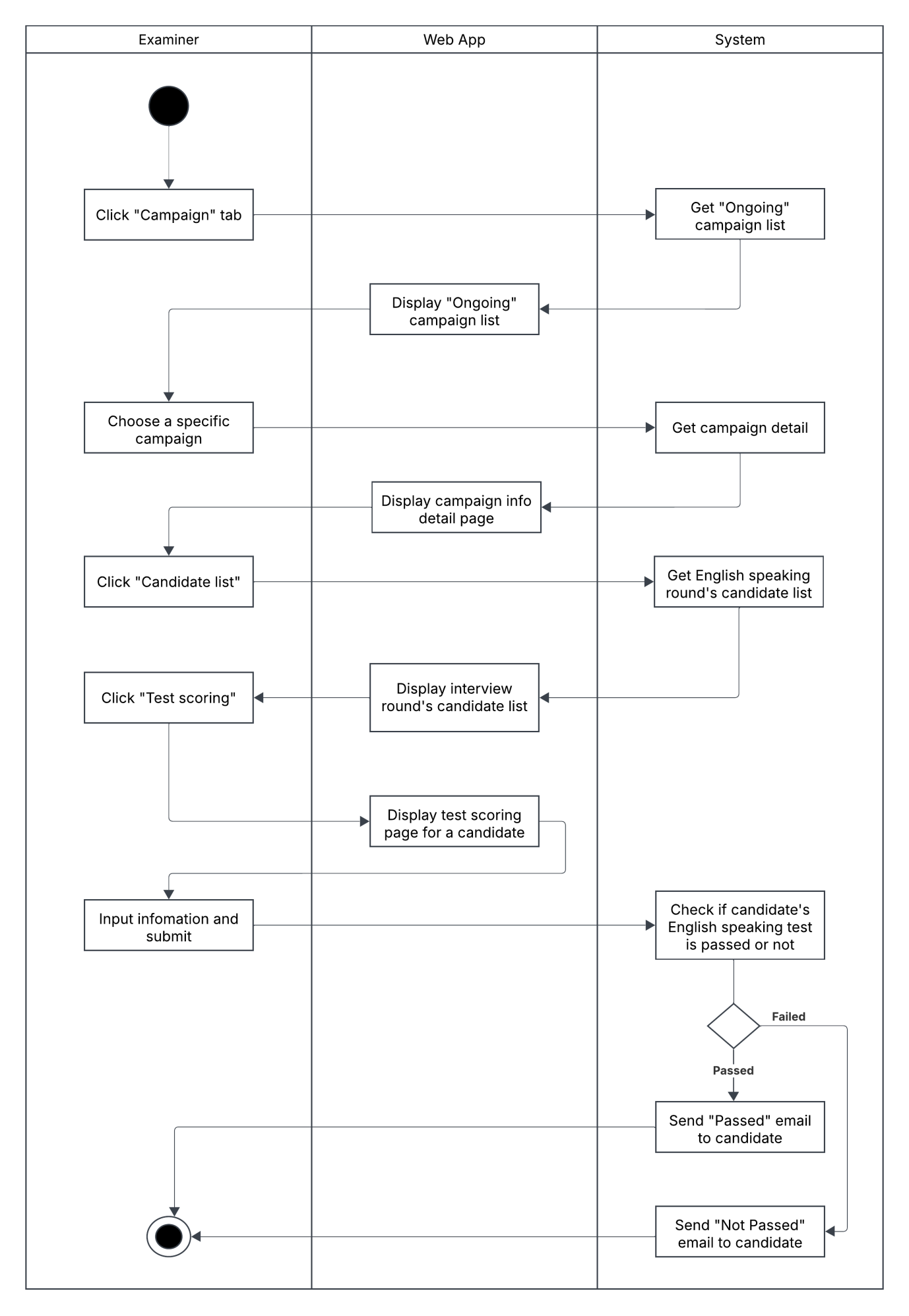
#### 1.3.18 Export Excel File For Post-Checking Activity Diagram



#### 1.3.19 Create Enquiry Request Activity Diagram



#### 1.3.20 English Speaking Test Scoring Activity Diagram

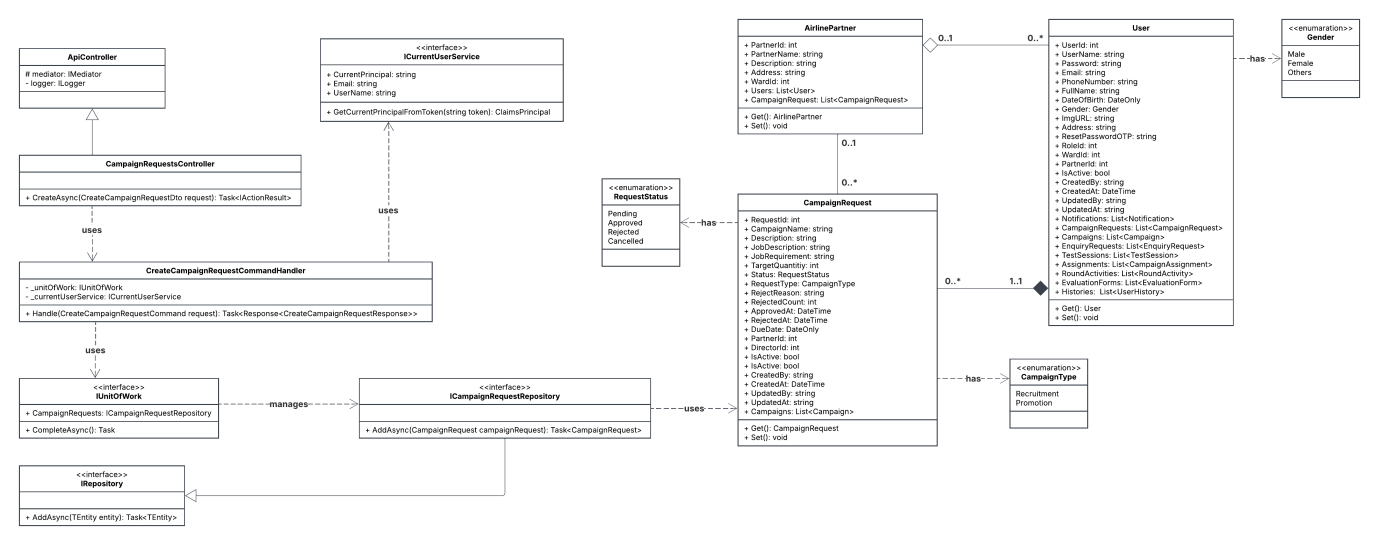


### 1.4 State Machine Diagrams

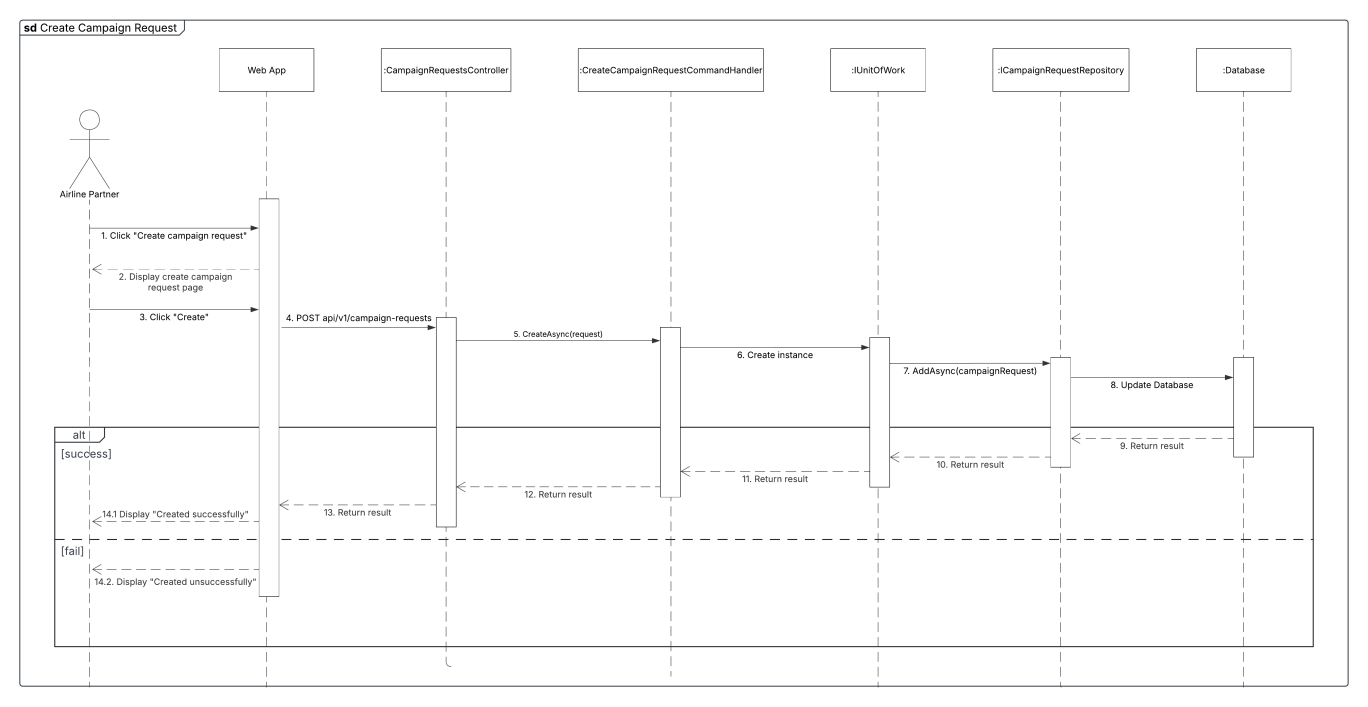
## 2. Detailed Design

### 2.1 Create Campaign Request

#### 2.1.1 Class Diagram

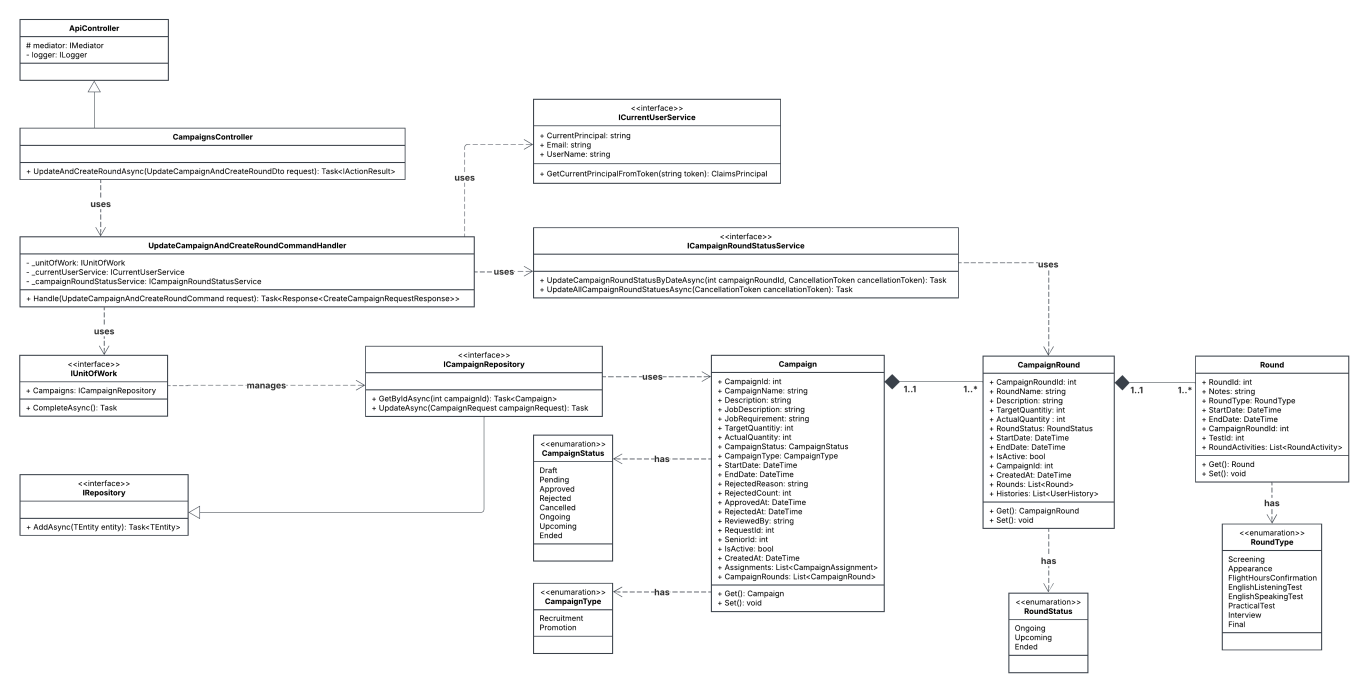


#### 2.1.2 Sequence Diagram

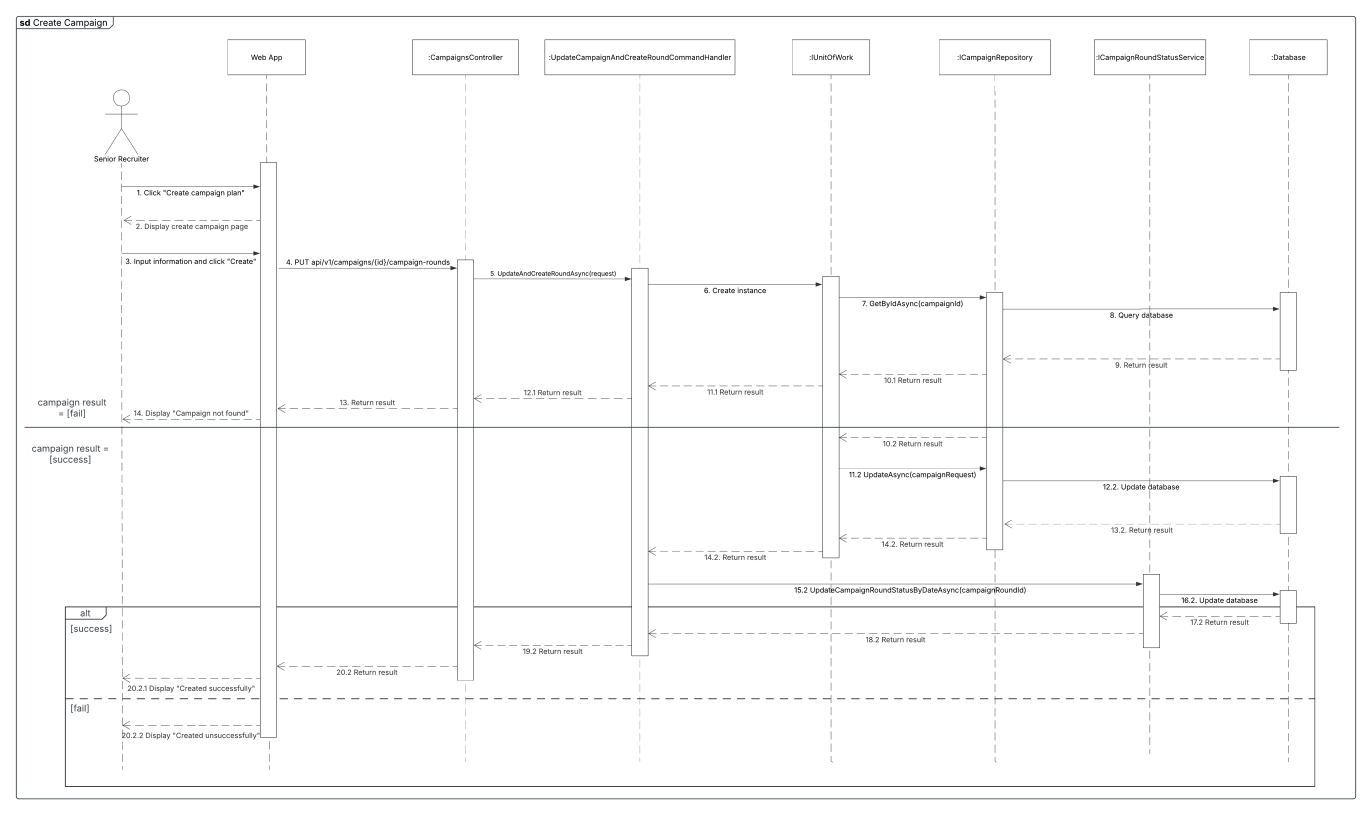


### 2.2 Create Campaign

#### 2.2.1 Class Diagram

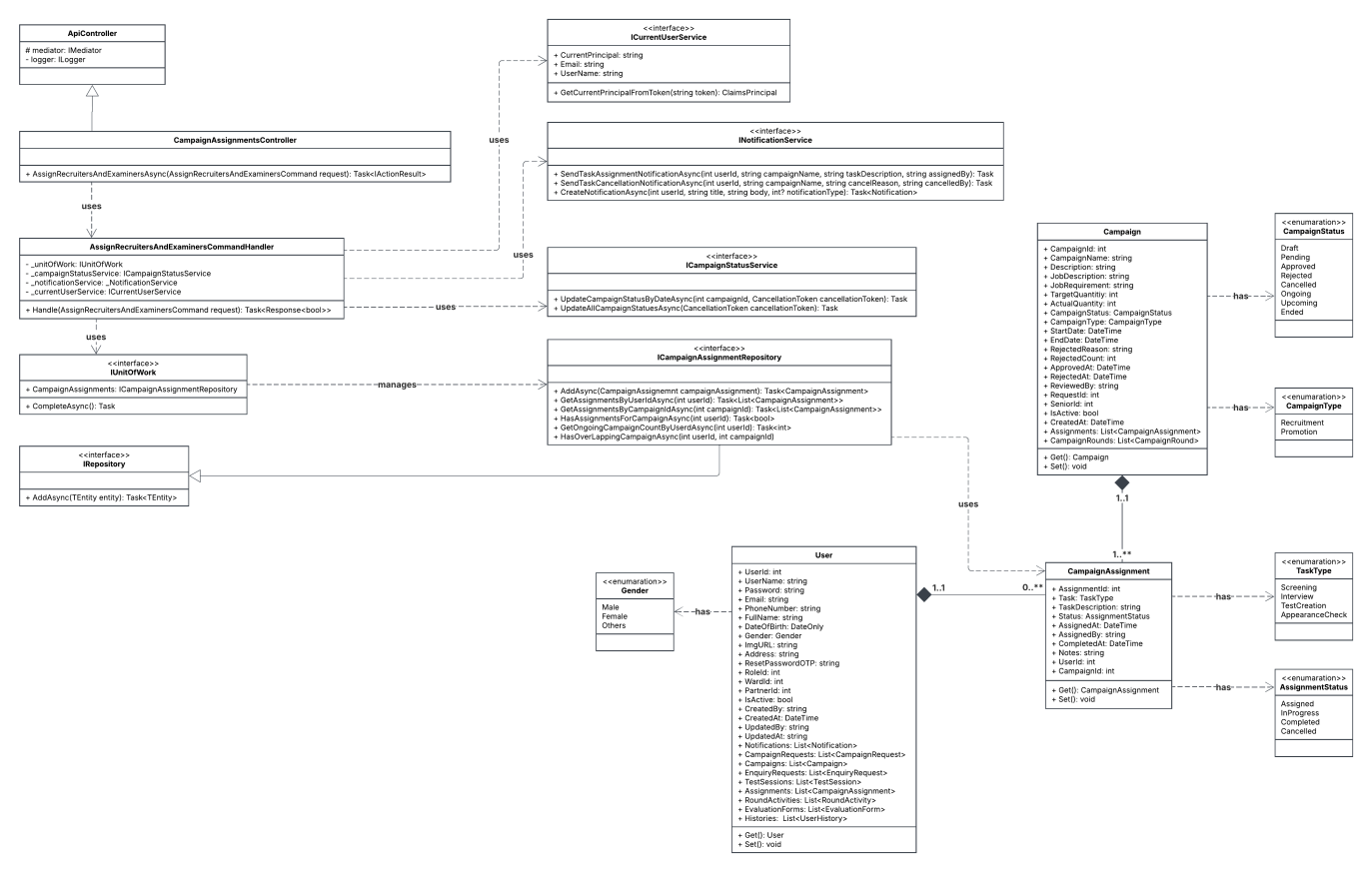


#### 2.2.1 Sequence Diagram

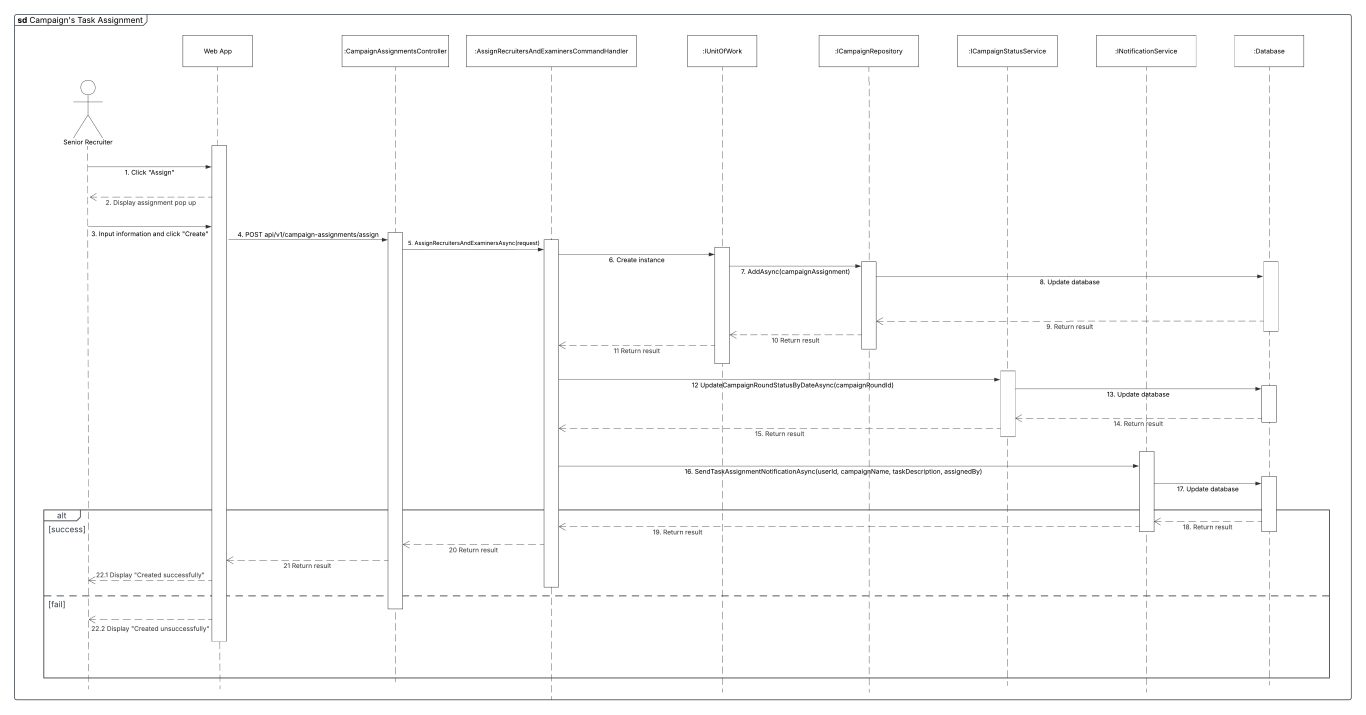


### 2.3 Campaign Task’s Assignment

#### 2.3.1 Class Diagram

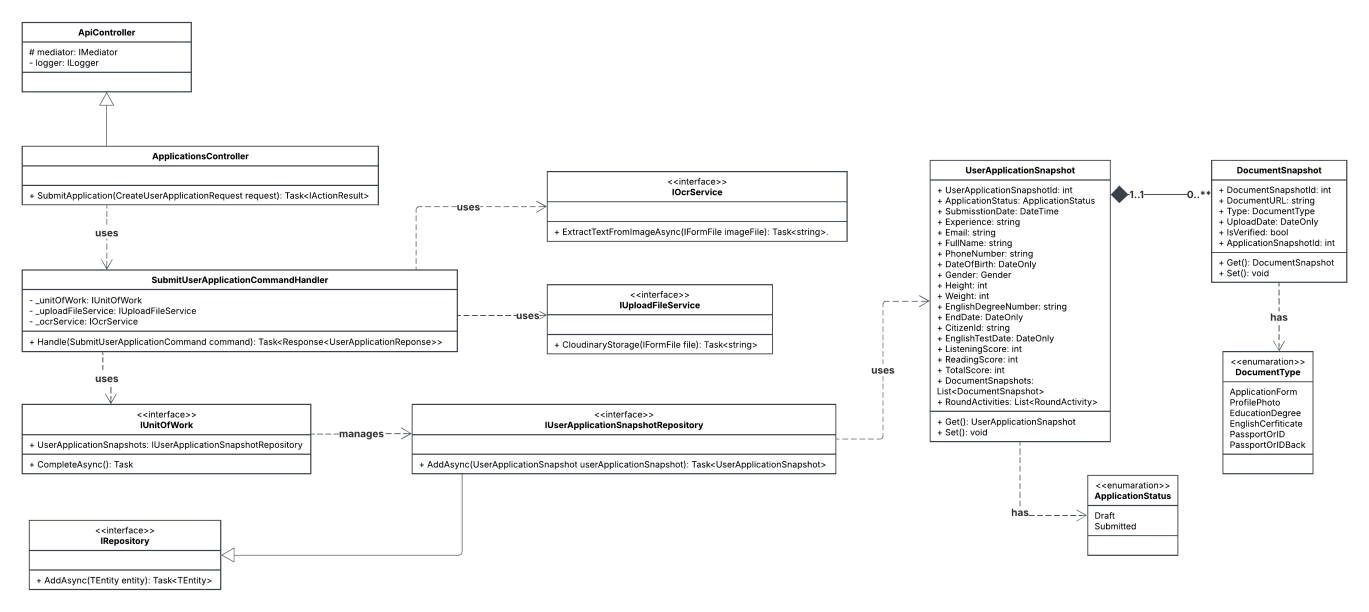


#### 2.3.1 Sequence Diagram

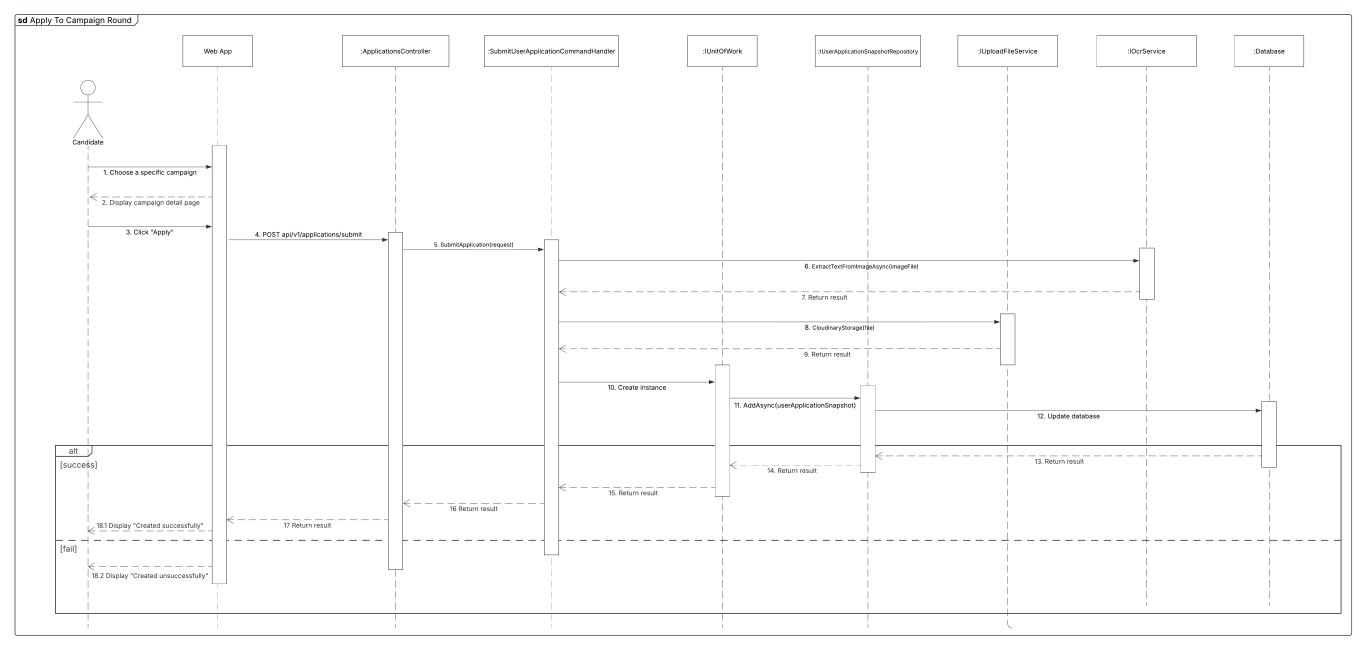


### 2.4 Apply To Campaign Round

#### 2.4.1 Class Diagram

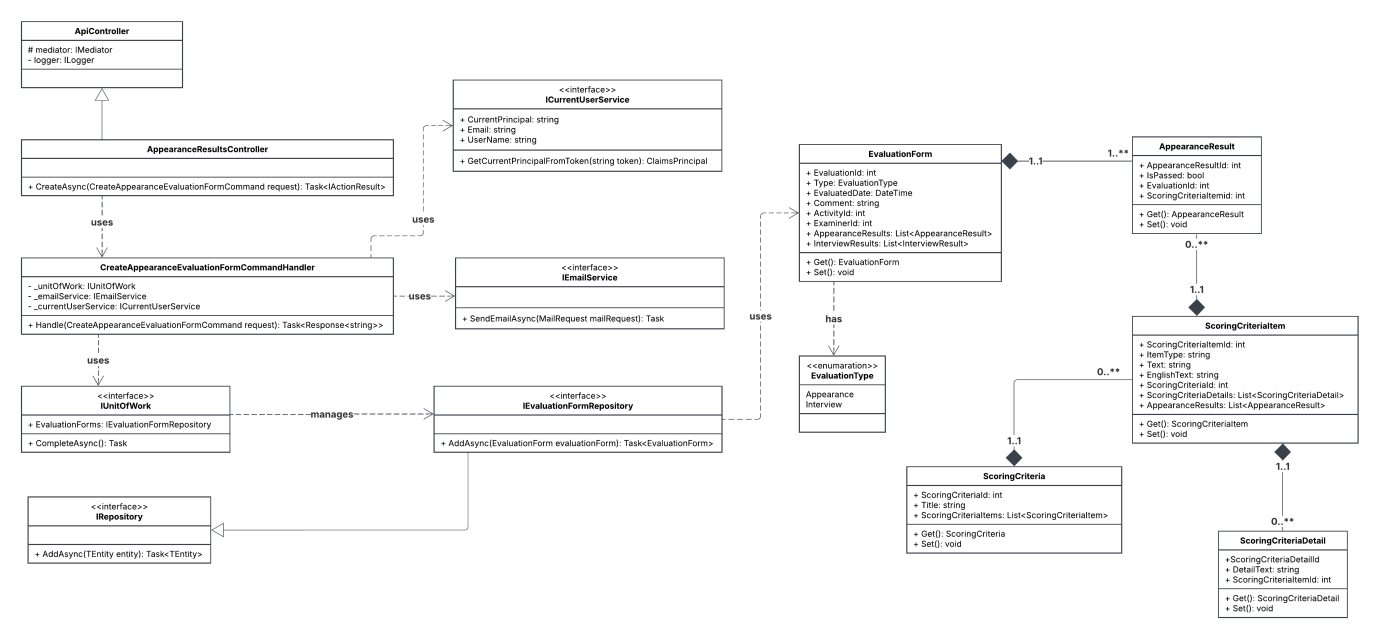


#### 2.4.1 Sequence Diagram

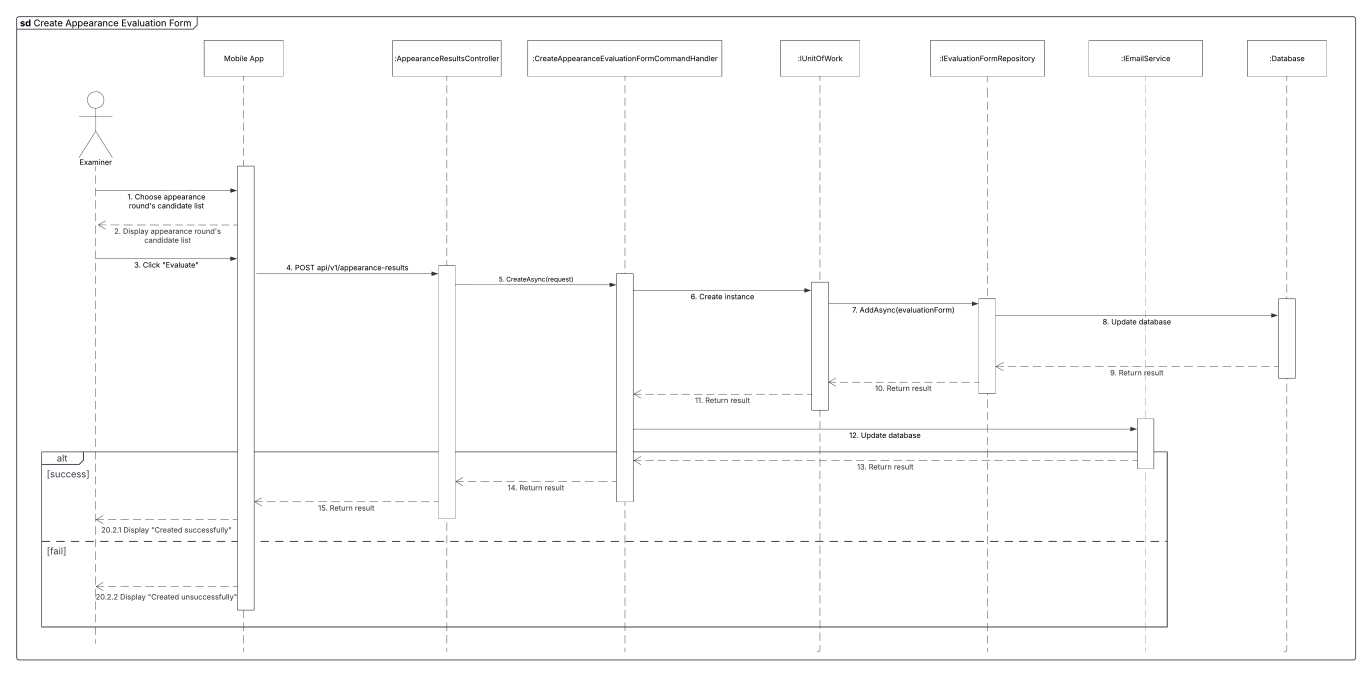


### 2.5 Create Appearance Evaluation Form

#### 2.5.1 Class Diagram

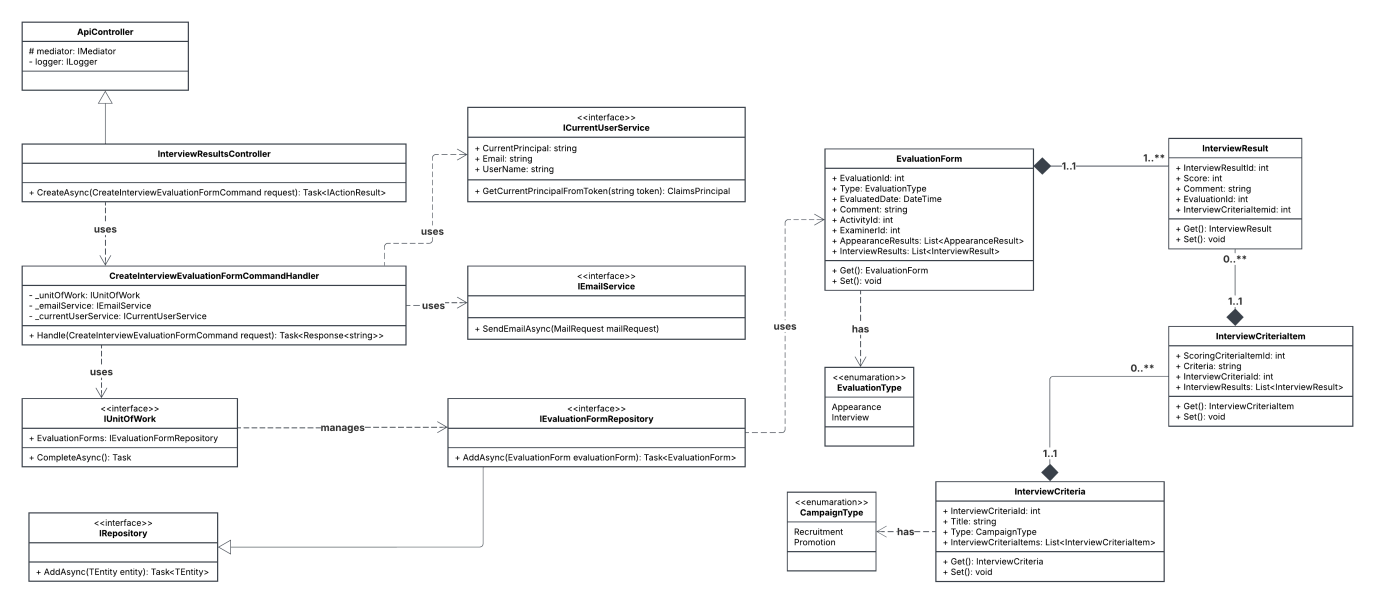


#### 2.5.1 Sequence Diagram

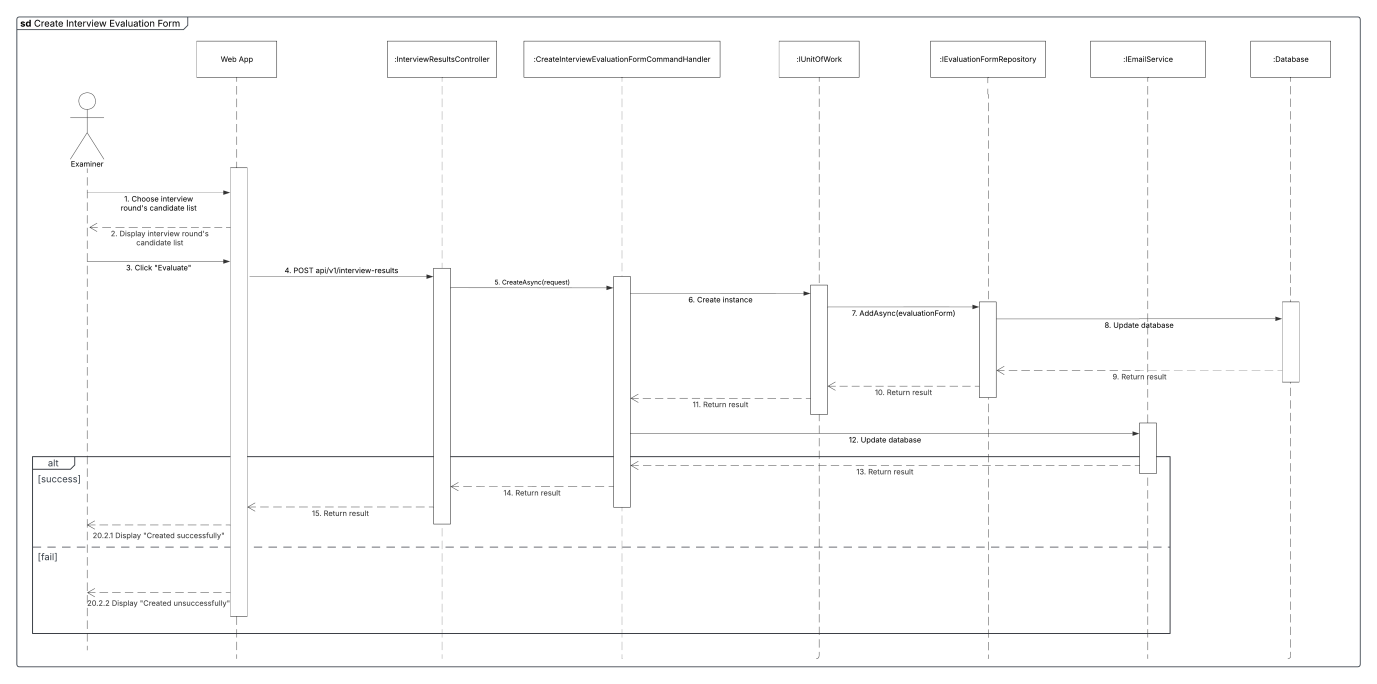


### 2.6 Create Interview Evaluation Form

#### 2.6.1 Class Diagram

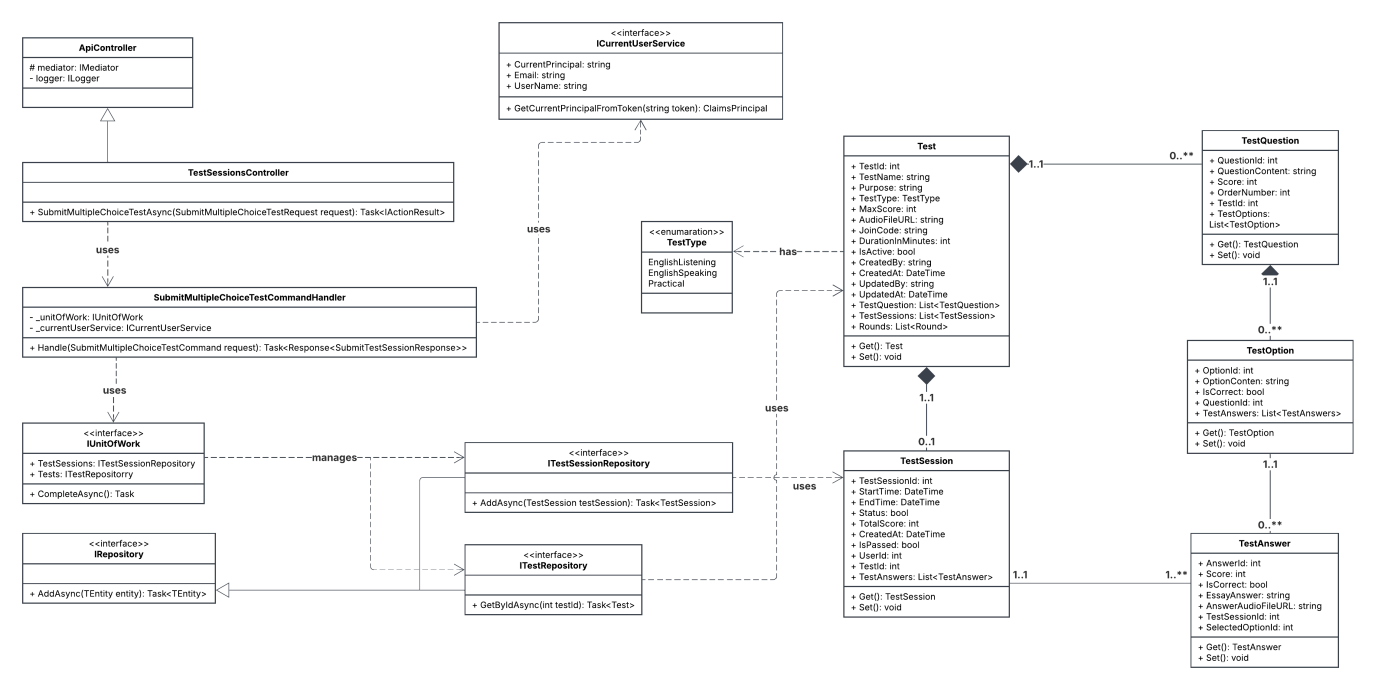


#### 2.6.1 Sequence Diagram

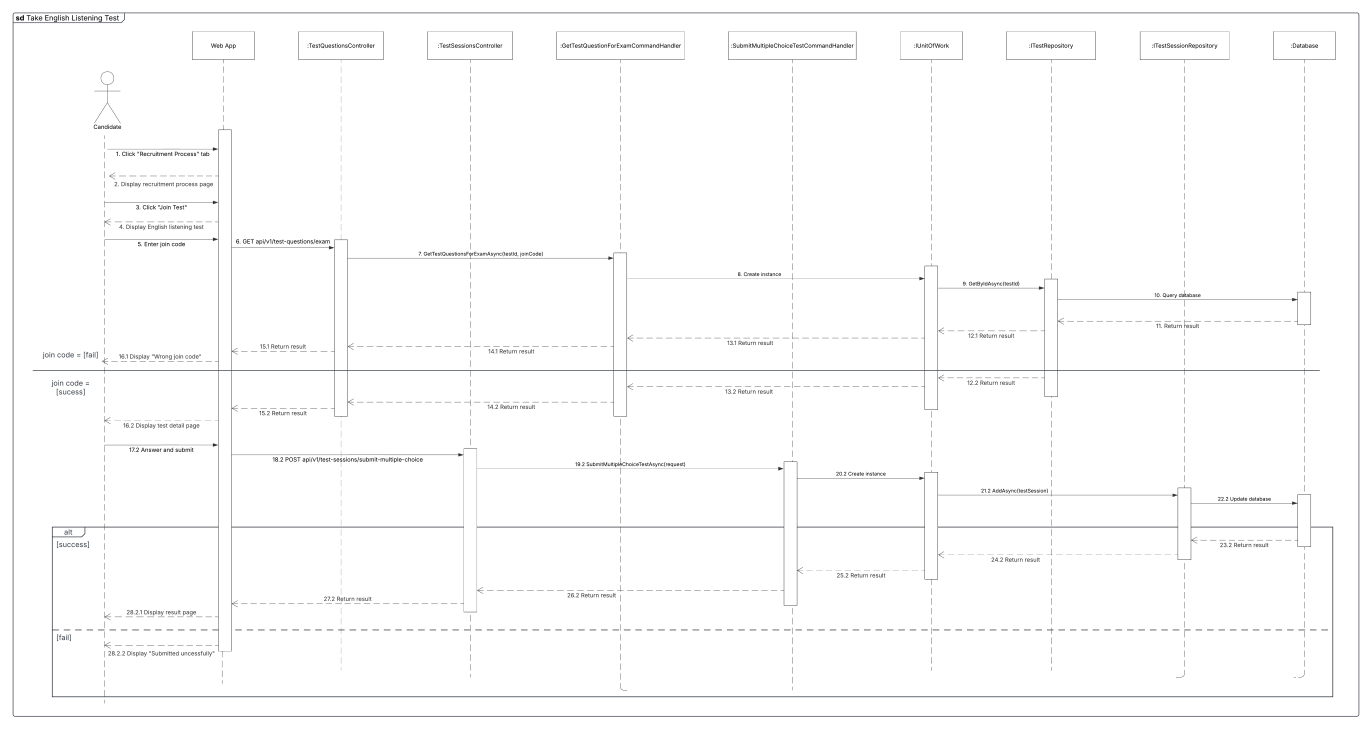


### 2.7 Take English Test

#### 2.7.1 Class Diagram

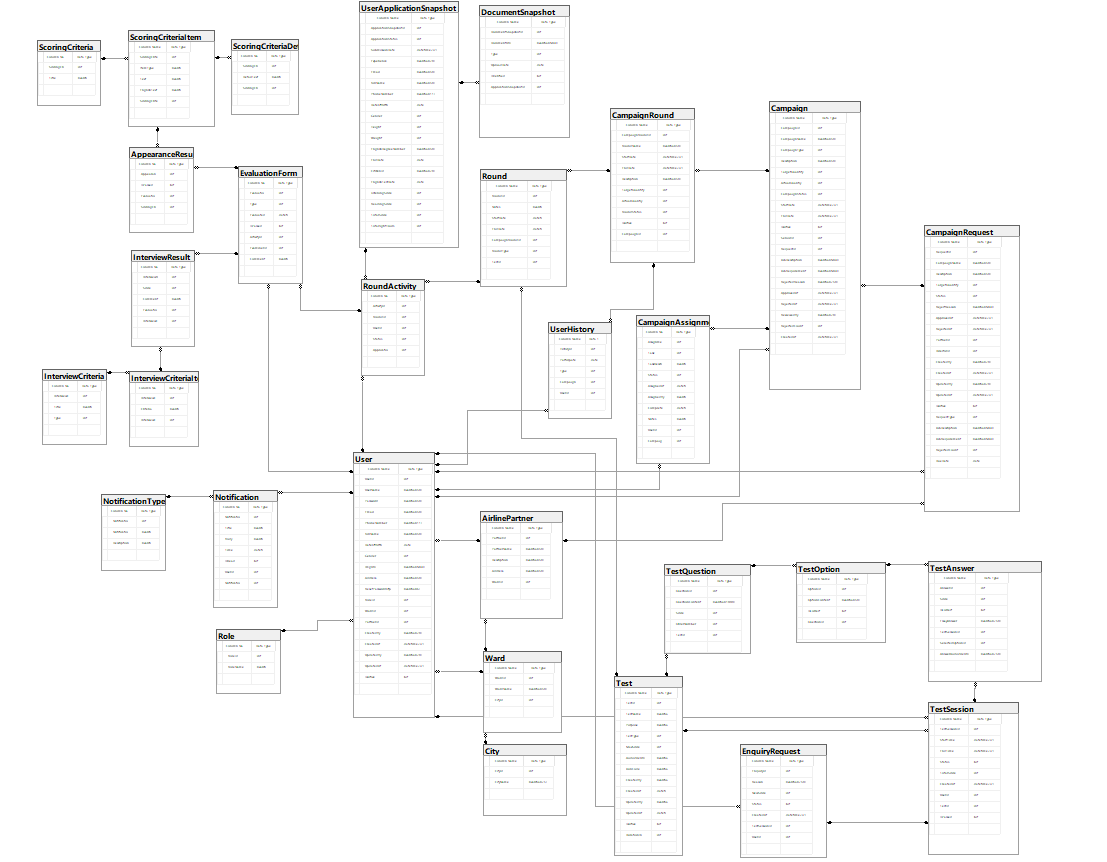


#### 2.7.1 Sequence Diagram



## 3. Database Design

### 3.1 Diagram



### 3.2 Tables Description

#### 3.2.1 AirlinePartner Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| PartnerId | Int | Unique identifier for the partner | Yes | Yes | PK |
| PartnerName | Nvarchar(50) | Name of the the partner | No | Yes | No |
| Description | Nvarchar(50) | Description of the partner | No | No | No |
| Address | Nvarchar(50) | Address of the partner | No | No | No |
| WardId | Int | Define the ward of the partner | No | No | FK |

#### 3.2.2 AppearanceResult Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| AppearanceResultId | Int | Unique identifier for the appearance result | Yes | Yes | PK |
| IsPassed | Bit | To check if this appearance result is passed or not | No | Yes | No |
| EvaluationId | Int | Define the evaluation form of the appearance result | No | No | FK |
| ScoringCriteriaItemId | Int | Define the scoring criteria item of the apperance result | No | No | FK |

#### 3.2.3 Campaign Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| CampaignId | Int | Unique identifier for the campaign | Yes | Yes | PK |
| CampaignName | Nvarchar(50) | Name of the campaign | No | Yes | No |
| CampaignType | Int | Type of the campaign | No | Yes | No |
| Description | Nvarchar(50) | Description of the campaign | No | No | No |
| TargetQuantity | Int | Target quantity of the campaign | No | Yes | No |
| ActualQuantity | Int | Actual Quantity of the campaign | No | Yes | No |
| CampaignStatus | Int | Status of the campaign | No | No | No |
| StartDate | Datetime2(7) | Define when the campaign starts | No | Yes | No |
| EndDate | Datetime2(7) | Define when the campaigns ends | No | Yes | No |
| IsActive | Bit | Check if the campaign is active or not | No | No | No |
| SeniorId | Int | Define senior recruiter of the campaign | No | No | FK |
| RequestId | Int | Define the campaign request of the campaign | No | No | FK |
| JobDescription | Nvarchar(MAX) | Job description of the campaign | No | No | No |
| JobRequirement | Nvarchar(MAX) | Job requirement of the campaign | No | No | No |
| RejectedReason | Nvarchar(250) | Rejected reason of the campaign | No | Yes | No |
| ApprovedAt | Datetime2(7) | Define when the campaign was approved | No | Yes | No |
| RejectedAt | Datetime2(7) | Define when the campaign was rejected | No | Yes | No |
| ReviewedBy | Nvarchar(20) | Define who reviewed the campaign | No | Yes | No |
| RejectedCount | Int | Rejected count of the campaign | No | No | No |
| CreatedAt | Datetime2(7) | Define when the campaign was created | No | Yes | No |

#### 3.2.4 CampaignAssignment Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| AssignmentId | Int | Unique identifier for the assignment | Yes | Yes | PK |
| Task | Int | Type of the task | No | Yes | No |
| TaskDescription | Nvarchar(100) | Description of the task | No | Yes | No |
| Status | Int | Status of the assignment | No | Yes | No |
| AssignedAt | Datetime2(7) | Define when the task was assigned | No | No | No |
| AssignedBy | Nvarchar(20) | Define who assigned the task | No | No | No |
| CompletedAt | Datetime2(7) | Define when the task was completed | No | No | No |
| Notes | Nvarchar(250) | Notes of the assignment | No | Yes | No |
| UserId | Int | Define the user of this assignment | No | Yes | FK |
| CampaignId | Int | Define the campaign of this assignment | No | Yes | FK |

#### 3.2.5 CampaignRequest Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| RequestId | Int | Unique identifier for the campaign request | Yes | Yes | PK |
| CampaignName | Nvarchar(50) | Name of the campaign request | No | Yes | No |
| Description | Nvarchar(50) | Description of the campaign request | No | Yes | No |
| TargetQuantity | Int | Target quantity of the campaign request | No | Yes | No |
| Status | Int | Status of the campaign request | No | Yes | No |
| RejectedReason | Nvarchar(MAX) | Rejected reason of the campaign request | No | Yes | No |
| ApprovedAt | Datetime2(7) | Define when the campaign request was approved | No | No | No |
| RejectedAt | Datetime2(7) | Define when the campaign request was rejected | No | No | No |
| PartnerId | Int | Define the partner of this campaign request | No | Yes | FK |
| DirectorId | Int | Define the director of this campaign request | No | Yes | FK |
| CreatedBy | Nvarchar(20) | Define who created the campaign request | No | No | No |
| CreatedAt | Datetime2(7) | Define when the campaign request was created | No | No | No |
| UpdatedBy | Nvarchar(20) | Define who updated the campaign request | No | No | No |
| UpdatedAt | Datetime2(7) | Define when the campaign request was updated | No | No | No |
| IsActive | Bit | Check if the campaign request is active or not | No | Yes | No |
| RequestType | Int | Type of the campaign request | No | No | No |
| JobDescription | Nvarchar(MAX) | Job description of the campaign request | No | Yes | No |
| JobRequirement | Nvarchar(MAX) | Job requirement of the campaign request | No | Yes | No |
| RejectedCount | Int | Rejected count of the campaign request | No | Yes | No |
| DueDate | Date | When the campaign request is due | No | No | No |

#### 3.2.6 CampaignRound Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| CampaignRoundId | Int | Unique identifier for the campaign round | Yes | Yes | PK |
| RoundName | Nvarchar(50) | Name of the campaign round | No | Yes | No |
| StartDate | Datetime2(7) | Define when the campaign round starts | No | No | No |
| EndDate | Datetime2(7) | Define when the campaign round ends | No | No | No |
| Description | Nvarchar(50) | Description of the campaign round | No | Yes | No |
| TargetQuantity | Int | Target quantity of the campaign round | No | Yes | No |
| ActualQuantity | Int | Actual quantity of the campaign round | No | Yes | No |
| RoundStatus | Int | Status of the campaign round | No | Yes | No |
| IsActive | Bit | Check if the campaign round is active or not | No | Yes | No |
| CampaignId | Int | Define the campaign of this campaign round | No | Yes | FK |

#### 3.2.7 City Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| CityId | Int | Unique identifier for the city | Yes | Yes | PK |
| CityName | Nvarchar(25) | Name of the city | No | Yes | No |

#### 3.2.8 DocumentSnapshot Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| DocumentSnapshotId | Int | Unique identifier for the document snapshot | Yes | Yes | PK |
| DocumentURL | Nvarchar(MAX) | Url of the document snapshot | No | Yes | No |
| Type | Int | Type of the document snapshot | No | Yes | No |
| UploadDate | Date | Define when the document snapshot was uploaded | No | Yes | No |
| IsVerified | Bit | Check if the document snapshot is verified or not | No | Yes | No |
| ApplicationSnapshotId | Int | Define the application snapshot of this document snapshot | No | Yes | FK |

#### 3.2.9 EnquiryRequest Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| EnquiryId | Int | Unique identifier for the enquiry request | Yes | Yes | PK |
| Reason | Nvarchar(250) | Reason for the enquiry request | No | Yes | No |
| NewScore | Int | New score for the test of this enquiry request | No | Yes | No |
| Status | Bit | Status of the enquiry request | No | Yes | No |
| CreatedAt | Datetime2(7) | Define when the enquiry request was created | No | No | No |
| TestSessionId | Int | Define the test session of this enquiry request | Yes | Yes | FK |
| UserId | Int | Define the user of this enquiry request | No | Yes | FK |

#### 3.2.10 EvaluationForm Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| EvaluationId | Int | Unique identifier for the evaluation form | Yes | Yes | PK |
| Type | Int | Type of the evaluation form | No | Yes | No |
| EvaluatedDate | Datetime2(7) | Define when the evaluation form was evaluated | No | No | No |
| IsPassed | Bit | Check if the evaluation form is passed or not | No | Yes | No |
| ActivityId | Int | Define the round activity of this evaluation form | No | Yes | FK |
| ExaminerId | Int | Define the examiner of this evaluation form | No | Yes | FK |
| Comment | Nvarchar(100) | Comment of this evaluation form | No | Yes | No |

#### 3.2.11 InterviewCriteria Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| InterviewCriteriaId | Int | Unique identifier for the partner | Yes | Yes | PK |
| Title | Nvarchar(50) | Name of the partner | No | Yes | No |
| Type | Int |  |  |  |  |

#### 3.2.12 InterviewCriteriaItem Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| InterviewCriteriaItemId | Int | Unique identifier for the partner | Yes | Yes | PK |
| Criteria | Nvarchar(250) | Name of the partner | No | Yes | No |
| InterviewCriteriaId | Int |  |  |  |  |

#### 3.2.13 InterviewResult Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| InterviewResultId | Int | Unique identifier for the interview result | Yes | Yes | PK |
| Score | Int | Score of the interview result | No | Yes | No |
| Comment | Nvarchar(50) | Comment of the interview result | No | Yes | No |
| EvaluationId | Int | Define the evaluation form of this intervew result | No | Yes | FK |
| InterviewCriteriaItemId | Int | Define the interview criteria item of this interview result | No | Yes | FK |

#### 3.2.14 Notification Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| NotificationId | Int | Unique identifier for the notification | Yes | Yes | PK |
| Title | Nvarchar(50) | Title of the notification | No | Yes | No |
| Body | Nvarchar(250) | Body of the notification | No | Yes | No |
| Time | Datetime2(7) | Define when the notification was created | No | No | No |
| IsRead | Bit | Check if the notification is read or not | No | Yes | No |
| UserId | Int | Define the user of this notification | No | Yes | FK |
| NotificationTypeId | Int | Define the notification type of this notificaiton | No | Yes | FK |

#### 3.2.15 NotificationType Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| NotificationTypeId | Int | Unique identifier for the notification type | Yes | Yes | PK |
| NotificationTypeName | Nvarchar(25) | Name of the notification type | No | Yes | No |
| Description | Nvarchar(50) | Description of the notification type | No | Yes | No |

#### 3.2.16 Role Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| RoleId | Int | Unique identifier for the role | Yes | Yes | PK |
| RoleName | Nvarchar(20) | Name of the role | No | Yes | No |

#### 3.2.17 Round Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| RoundId | Int | Unique identifier for the round | Yes | Yes | PK |
| Notes | Nvarchar(50) | Notes of the round | No | Yes | No |
| StartDate | Datetime2(7) | Define when the round starts | No | No | No |
| EndDate | Datetim2(7) | Define when the round ends | No | No | No |
| CampaignRoundId | Int | Define the campaign round of this round | No | Yes | FK |
| RoundType | Int | Type of the round | No | Yes | No |
| TestId | Int | Define the test of this round | No | Yes | No |

#### 3.2.18 RoundActivity Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| ActivityId | Int | Unique identifier for the round activity | Yes | Yes | PK |
| RoundId | Int | Define the round of this round activity | No | Yes | FK |
| UserId | Int | Define the user of this round activity | No | Yes | FK |
| Status | Int | Status of the round activity | No | Yes | No |
| ApplicationSnapshotId | Int | Define the user application snapshot of this round activity | No | Yes | FK |

#### 3.2.19 ScoringCriteria Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| ScoringCriteriaId | Int | Unique identifier for the scoring criteria | Yes | Yes | PK |
| Title | Nvarchar(50) | Title of the scoring criteria | No | Yes | No |

#### 3.2.20 ScoringCriteriaDetail Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| ScoringCriteriaDetailId | Int | Unique identifier for the scoring criteria detail | Yes | Yes | PK |
| DetailText | Nvarchar(100) | Detail text of the scoring criteria detail | No | Yes | No |
| ScoringCriteriaItemId | Int | Define the scoring criteria item of this scoring criteria detail | No | Yes | FK |

#### 3.2.21 ScoringCriteriaItem Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| ScoringCriteriaItemId | Int | Unique identifier for the scoring criteria item | Yes | Yes | PK |
| ItemType | Nvarchar(50) | Type of the scoring criteria item | No | Yes | No |
| Text | Nvarchar(250) | Text of the scoring criteria item | No | Yes | No |
| EnglishText | Nvarchar(250) | English text of the scoring criteria item | No | Yes | No |
| ScoringCriteriaId | Int | Define the scoring criteria of this scoring criteria item | No | Yes | FK |

#### 3.2.22 Test Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| TestId | Int | Unique identifier for the test | Yes | Yes | PK |
| TestName | Nvarchar(100) | Name of the test | No | Yes | No |
| Purpose | Nvarchar(100) | Purpose of the test | No | Yes | No |
| TestType | Int | Type of the test | No | Yes | No |
| MaxScore | Int | Max score of the text | No | Yes | No |
| AudioFileURL | Nvarchar(MAX) | Audio file Url of the test | No | No | No |
| JoinCode | Nvarchar(20) | Join code of the test | No | Yes | No |
| CreatedBy | Nvarchar(20) | Define who created the test | No | No | No |
| CreatedAt | Datetime2(7) | Define when the test was created | No | No | No |
| UpdatedBy | Nvarchar(20) | Define who updated the test | No | No | No |
| UpdatedAt | Datetime2(7) | Define when the test was updated | No | No | No |
| IsActive | Bit | Check if the test is active or not | No | Yes | No |
| DurationInMinutes | Int | Duration in minutes of the test | No | Yes | No |

#### 3.2.23 TestAnswer Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| AnswerId | Int | Unique identifier for the test answer | Yes | Yes | PK |
| Score | Int | Score of the test answer | No | Yes | No |
| IsCorrect | Bit | Check if the test answer is correct or not | No | Yes | No |
| EssayAnswer | Nvarchar(250) | Essay answer of the test answer | No | Yes | No |
| TestSessionId | Int | Define the test session of this test answer | No | Yes | FK |
| SelectedOptionId | Int | Define the selected option of this test answer | No | Yes | FK |
| AnswerAudioFileURL | Nvarchar(MAX) | Answer audio file Url of this test answer | No | No | No |

#### 3.2.24 TestOption Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| OptionId | Int | Unique identifier for the test option | Yes | Yes | PK |
| OptionContent | Nvarchar(50) | Option content of the test option | No | Yes | No |
| IsCorrect | Bit | Check if the test option is correct or not | No | Yes | No |
| QuestionId | Int | Define the test question of this test option | No | Yes | FK |

#### 3.2.25 TestQuestion Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| QuestionId | Int | Unique identifier for the test question | Yes | Yes | PK |
| QuestionContent | Nvarchar(1000) | Question content of this test question | No | Yes | No |
| Score | Int | Score of this test question | No | Yes | No |
| OrderNumber | Int | Order number of this test question | No | Yes | No |
| TestId | Int | Define the test of this test question | No | Yes | FK |

#### 3.2.26 TestSession Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| TestSessionId | Int | Unique identifier for the test session | Yes | Yes | PK |
| StartTime | Datetime2(7) | Define when the test session starts | No | No | No |
| EndTime | Datetime2(7) | Define when the test session ends | No | No | No |
| Status | Bit | Status of the test session | No | Yes | No |
| TotalScore | Int | Total score of the test session | No | Yes | No |
| CreatedAt | Datetime2(7) | Define when the test session was created | No | No | No |
| UserId | Int | Define the user of this test session | No | Yes | FK |
| TestId | Int | Define the test of this test session | No | Yes | FK |
| IsPassed | Bit | Check if the test session is passed or not | No | Yes | No |

#### 3.2.27 User Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| UserId | Int | Unique identifier for the user | Yes | Yes | PK |
| UserName | Nvarchar(50) | Username of the user | No | Yes | No |
| Password | Nvarchar(50) | Password of the user | No | Yes | No |
| Email | Nvarchar(50) | Email of the user | No | Yes | No |
| PhoneNumber | Nvarchar(11) | Phone number of the user | No | Yes | No |
| FullName | Nvarchar(50) | Full name of the user | No | Yes | No |
| DateOfBirth | Date | Date of birth of the user | No | No | No |
| Gender | Int | Gender of the user | No | Yes | No |
| ImgURL | Nvarchar(MAX) | Image Url of the user | No | Yes | No |
| Address | Nvarchar(50) | Address of the user | No | No | No |
| ResetPasswordOTP | Nvarchar(6) | Reset password OTP of the user | No | No | No |
| RoleId | Int | Define the role of this user | No | Yes | FK |
| WardId | Int | Define the ward of this user | No | Yes | FK |
| PartnerId | Int | Define the partner of this user | No | No | FK |
| CreatedBy | Nvarchar(20) | Define who created the user | No | No | No |
| CreatedAt | Datetime2(7) | Define when the user was created | No | No | No |
| UpdatedBy | Nvarchar(20) | Define who updated the user | No | No | No |
| UpdatedAt | Datetime2(7) | Define when the user was updated | No | No | No |
| IsActive | Bit | Check if the user is active or not | No | Yes | No |

#### 3.2.28 UserApplicationSnapshot Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| ApplicationSnapshotId | Int | Unique identifier for the user’s application snapshot | Yes | Yes | PK |
| ApplicationStatus | Int | Status of the user’s application | No | Yes | No |
| SubmissionDate | Datetime2(7) | Submitted date of the user’s application | No | No | No |
| Experience | Nvarchar(20) | Experience of the user’s application | No | No | No |
| Email | Nvarchar(50) | Email of the user | No | Yes | No |
| FullName | Nvarchar(50) | Full name of the user | No | Yes | No |
| PhoneNumber | Nvarchar(11) | Phone number of the user | No | Yes | No |
| DateOfBirth | Date | Date of birth of the user | No | No | No |
| Gender | Int | Gender of the user | No | Yes | No |
| Height | Int | Height of the user’s application | No | Yes | No |
| Weight | Int | Weight of the user’s application | No | Yes | No |
| EnglishDegreeNumber | Nvarchar(50) | English degree number of the application | Yes | Yes | No |
| EndDate | Date | Define when the English degree ends | No | No | No |
| CitizenId | Nvarchar(20) | Social security number of the user’s application | Yes | Yes | No |
| EnglishTestDate | Date | Define when the user took the English degree | No | No | No |
| ListeningScore | Int | Listening score of the user’s application | No | Yes | No |
| ReadingScore | Int | Reading score of the user’s application | No | Yes | No |
| TotalScore | Int | Total score of the user’s application | No | Yes | No |
| TotalFlightHours | Int | Total Flight Hours of the user’s application | No | Yes | No |

#### 3.2.29 UserHistory Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| HistoryId | Int | Unique identifier for the user history | Yes | Yes | PK |
| ParticipatedDate | Date | Define when the user participated in the campaign round | No | Yes | No |
| Type | Int | Type of the user history | No | Yes | No |
| CampaignRoundId | Int | Define the campaign round of this user history | No | Yes | FK |
| UserId | Int | Define the user of this user history | No | Yes | FK |

#### 3.2.30 Ward Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **Type** | **Description** | **Unique** | **Not Null** | **PK/FK** |
| WardId | Int | Unique identifier for the ward | Yes | Yes | PK |
| WardName | Nvarchar(50) | Name of the ward | No | Yes | No |
| CityId | Int | Define the city of this ward | No | Yes | FK |

# V. Software Testing Documentation

## 1. Scope of Testing

## 2. Test Strategy

### 2.1 Testing Types

### 2.2 Test Levels

### 2.3 Supporting Tools

## 3. Test Plan

### 3.1 Human Resources

### 3.2 Test Environment

### 3.3 Test Milestones

## 4. Test Cases

## 5. Test Reports

# VI. Release Package & User Guides

## 1. Deliverable Package

| **No.** | **Deliverable Item** | **Description** |
| --- | --- | --- |
| 1 | Schedule/Task Tracking |  |
| 2 | Project Backlog |  |
| 3 | Source Codes |  |
| 4 | Database Script(s) |  |
| 5 | Final Report Document |  |
| 6 | Test Cases Document |  |
| 7 | Defects List |  |
| 8 | Issues List |  |
| 9 | Slide |  |

## 2. Installation Guides

### 2.1 System Requirements

### 2.2 Installation Instruction

## 3. User Manual

### 3.1 Overview

### 3.2 Workflow 1

### 3.3 Workflow 2

…