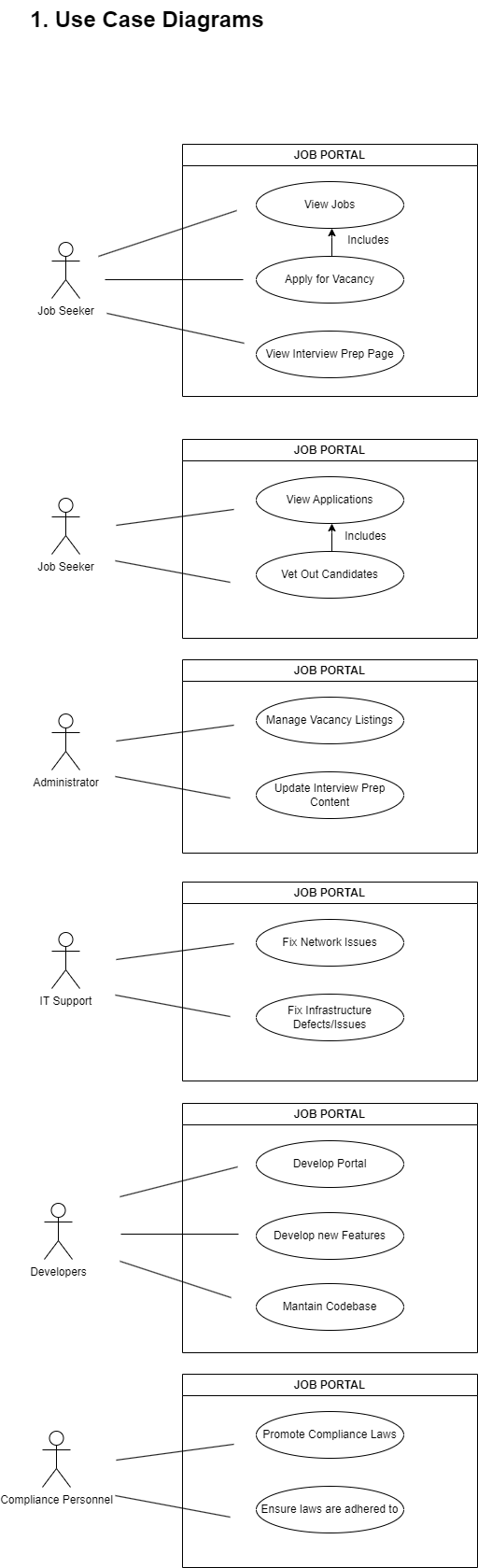
**Software Engineering: Assignment 5**

**Use Case Modeling and Test Case Development for the System**

1. **Use Case Diagrams**

****

**3.Use Case Specifications**

**Use Case:** *View Jobs*

**Actor:** Job seeker

**Precondition:** No tangible preconditions apart from the job seeker wanting to apply for any available opportunities.

**Post condition:** No post conditions identified.

**Basic Flow:**

1. Job seeker clicks on site on their browser
2. Job seeker views listed vacancies.

**Use Case**: *Apply For Vacancy*

**Actor:** Job seeker

**Precondition:** The job seeker must have viewed available jobs in order to apply.

**Post condition:** The system stores the application.

**Basic Flow:**

1. Job seeker views all vacancies available
2. Job seeker clicks on vacancy of interest
3. Job seeker views job details.
4. Job seeker clicks on apply.
5. Job seeker submits required documents.

**Use Case:** *View Applications*

**Actor:**HR personnel

**Precondition:** There must have been applications submitted.

**Post condition:** The HR personnel is vets out relevant applications

**Basic Flow:**

1. HR personnel accesses system
2. HR personnel accesses applications
3. HR personnel is able to see applicant info and vet out relevant candidates.

**Use Case:** *Manage Vacancies*

**Actor:**System Administrator

**Precondition:** There must be loaded vacancies on the system.

**Post condition:** Vacancy must be added, edited or removed.

**Basic Flow:**

1. System admin accesses the system
2. System admin adds, edits or removes vacancy from the system.

**Use Case:** *Develop Portal*

**Actor:**Software Developers

**Precondition:** All system requirements must be properly specified and not ambiguous.

**Post condition:** The website will be fully functional, and requirements met.

**Basic Flow:**

1. Planning
2. Analysis
3. Design
4. Develop
5. Implement/Deploy
6. Support/Maintain

**Use Case:** *Maintain codebase*

**Actor:** Software Developers

**Precondition:** Code has to be readable, scalable and understandable in order for maintenance to be possible. Correct framework and language is adhered to.

**Post condition:** Code meets maintenance requirements and standards.

**Basic Flow:**

1. Gather maintenance requirements
2. Follow correct tools, framework and languages
3. Perform maintenance on code.

**Use Case:** *Fix Network Issues*

**Actor:** IT Support

**Precondition:** Support personnel gather proper information on the issue, and what might’ve caused it.

**Post condition:** Network problem is fixed and problem scenario is tested again to confirm.

**Basic Flow**

1. Call is logged with IT Support
2. IT Support analyze the problem and the probable cause
3. IT Support troubleshoots issue
4. IT Support fixes the issue

**Use Case:** *Fix Infrastructure*

**Actor:** IT Support

**Precondition:** ITSupport personnel gather proper information on the issue, and what might be the cause.

**Post condition:** Infrastructure component is fixed/replaced, tested to confirm.

**Basic Flow**

1. Call is logged with IT Support
2. Problem is analyzed and cause is identified.
3. Infrastructure problem is fixed.
4. **Test Case Development**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Requirement** | **Steps** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| **TC-001** | Job Seeker views available jobs | Performance | 1.Navigate to website 2.View Jobs | All available jobs are shown on the screen | **..** | **..** |
| **TC-002** | Job Seeker applies for a vacancy | Usability | 1.Click on vacancy  2.Enter/Upload necessary info  3. Click apply | ‘Application Successful’ message displayed. Application sent through. | **..** | **..** |
| **TC-003** | HR views applications sent through | Security | 1.Navigate to applications section  2.View application information. | All applicant information is shown. | **..** | **..** |
| **TC-004** | Develop portal | Scalability/Security | Implement SDLC | System meets all requirements | **..** | **..** |
| **TC-005** | Develop new features | Scalability/Security | Implement SDLC | New features meet all requirements | **..** | **..** |
| **TC-006** | Fix network issues | Availability | 1.Identify problem 2.Analyse problem  3.Troubleshoot issue  4.Attend to problem | Network issue is fixed. | **..** | **..** |
| **TC-007** | Update Interview Preparation Section | Performance | 1.User navigates to interview preparation section  2.User views interview prep section. | Job Seeker/User is able to read content on interview preparation, and able to access external links. | **..** | **..** |
| **TC-008** | Compliance oversees application and selection process | Security | 1.Receive access to system  2.Verify processes | Compliance is able to oversee and verify application process. | **..** | **..** |

1. **Reflection**

The challenges I met with translating requirements to use cases were defining the actual flow of each use case for the use case specifications for some use cases. Some were not previously defined to pure detail, so it was a challenge to actually break them down. That challenge assisted me in knowing the use case, defining and understanding what is actually required even better. An example would be the IT Support fixing attending to a network issue. Defining the **basic flow** reminded me that it is more than just fixing/attending to the issue. A call has to be logged with IT, the issue needs to be analyzed, troubleshoot, then only after troubleshooting will the issue be fixed.

Another one was defining the preconditions for use cases that did not have any. I tried to find preconditions only to realize that some use cases do not actually have dependencies and do not need other states/actions in order to be.