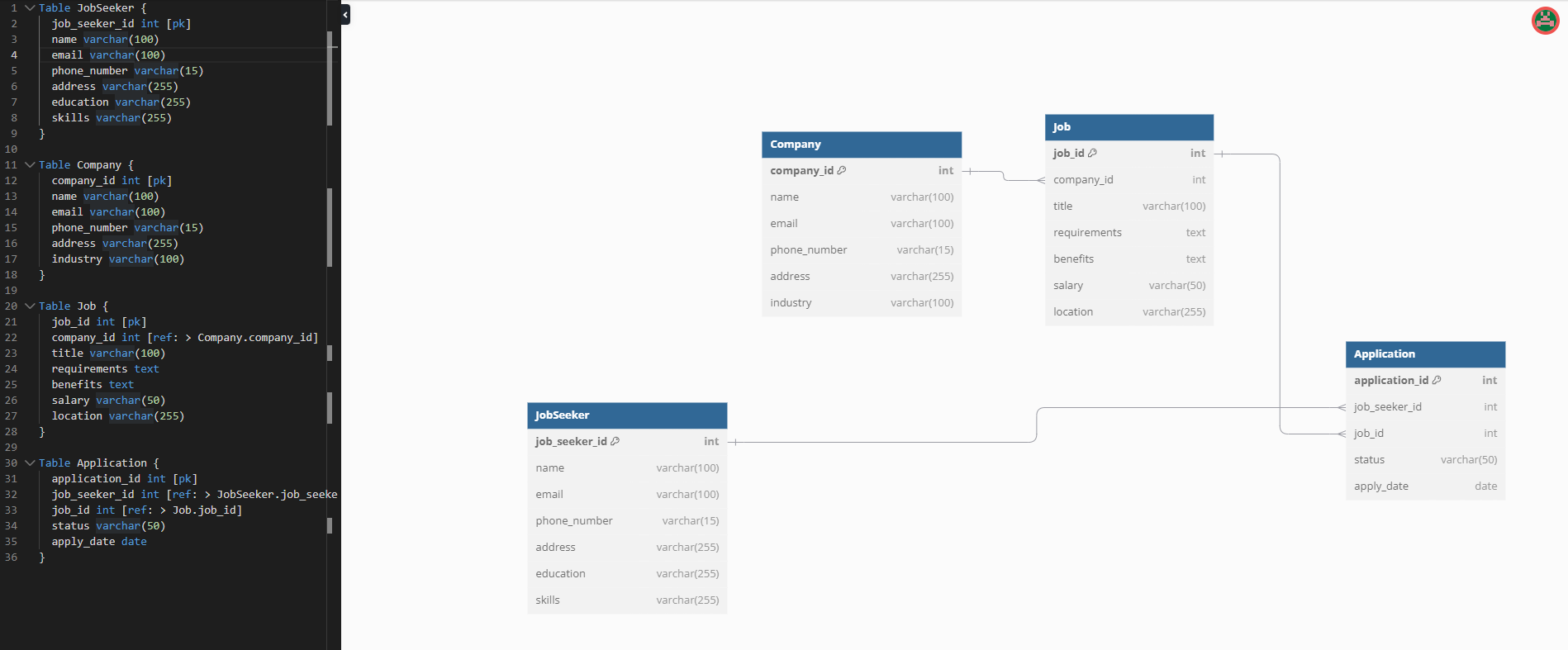
Computing Project

1.Design



2.Project process

**Các Nhánh Trong Git và Cách Dùng**

Trong dự án phát triển phần mềm, Technical Manager thường dùng 5 nhánh chính:

1. **Main**: Nhánh chính chứa mã đã kiểm tra, sẵn sàng cho sản xuất.
2. **Develop**: Nhánh làm việc chính, nơi tất cả mã của tính năng mới được tích hợp.
3. **Feature Branches**: Tạo từ develop để làm các tính năng mới, giúp làm việc độc lập mà không ảnh hưởng mã nguồn chính.
4. **Release Branches**: Chuẩn bị mã cho bản phát hành cuối, bao gồm kiểm tra và sửa lỗi.
5. **Hotfix Branches**: Sửa lỗi khẩn cấp trong sản phẩm và merge lại vào cả main và develop.

### **Các Bước Phát Triển Mô-đun Mới**

1. **Phân Tích Yêu Cầu**: Hiểu rõ chức năng cần làm.
2. **Lên Kế Hoạch**: Chia nhỏ module, xác định ai làm gì, và thời gian hoàn thành.
3. **Thiết Lập Môi Trường**: Cài đặt đủ công cụ và tạo nhánh feature từ develop.
4. **Thiết Kế**: Xác định cấu trúc và tương tác của module với hệ thống.
5. **Phát Triển Mã**: Code và commit thường xuyên.
6. **Kiểm Thử Đơn Vị**: Đảm bảo các phần chính hoạt động đúng.
7. **Kiểm Thử Tích Hợp**: Đưa mã vào develop và kiểm tra kết hợp với hệ thống.
8. **Code Review**: Gửi PR và sửa theo phản hồi.
9. **Kiểm Thử Cuối**: Chạy các test chức năng và hiệu năng nếu cần.
10. **Chuẩn Bị Phát Hành**: Merge vào nhánh release, kiểm tra lần cuối.
11. **Triển Khai**: Merge vào main, triển khai và giám sát.

3.Testing

### **Các Loại Chiến Lược Thử Nghiệm**

Trong phát triển phần mềm, có nhiều chiến lược thử nghiệm chính, bao gồm:

1. **Kiểm thử Đơn Vị (Unit Testing)**: Kiểm tra từng thành phần nhỏ nhất (hàm, class) để đảm bảo chúng hoạt động đúng.
2. **Kiểm thử Tích Hợp (Integration Testing)**: Kiểm tra các thành phần khi chúng được tích hợp với nhau, xem chúng có hoạt động đúng không.
3. **Kiểm thử Hệ Thống (System Testing)**: Đánh giá toàn bộ hệ thống, xem nó có hoạt động đúng với yêu cầu không.
4. **Kiểm thử Chấp Nhận (Acceptance Testing)**: Đảm bảo phần mềm phù hợp để phát hành, tức là đáp ứng đầy đủ yêu cầu người dùng.

### **Kiểm Thử Đơn Vị Để Làm Gì?**

Kiểm thử đơn vị tập trung vào từng phần nhỏ của mã (như hàm hoặc module) để chắc chắn chúng hoạt động như dự kiến. Nó giúp phát hiện lỗi ngay từ đầu, giảm rủi ro sau này khi tích hợp các phần lại với nhau.

### **Kiểm Thử Tích Hợp Để Làm Gì?**

Kiểm thử tích hợp được dùng để kiểm tra cách các phần trong hệ thống kết nối và làm việc với nhau. Nó đảm bảo rằng khi kết hợp lại, chúng vẫn hoạt động đúng mà không có lỗi tương tác.

### **Các Bước Kiểm Tra Một Mô-đun Mới**

Nếu có một module mới để kiểm tra, quy trình sẽ thường bao gồm:

1. **Hiểu Yêu Cầu**: Đọc tài liệu hoặc hỏi team xem module này cần làm gì.
2. **Lên Kế Hoạch Kiểm Thử**: Xác định các loại kiểm thử nào cần thực hiện (đơn vị, tích hợp, hệ thống).
3. **Thiết Kế Test Cases**: Viết các kịch bản kiểm thử cụ thể để đảm bảo không bỏ sót tình huống nào.
4. **Thực Hiện Kiểm Thử Đơn Vị**: Kiểm tra các phần nhỏ nhất của module.
5. **Thực Hiện Kiểm Thử Tích Hợp**: Kiểm tra xem module hoạt động thế nào khi kết hợp với các phần khác.
6. **Kiểm Thử Hệ Thống**: Chạy thử nghiệm toàn bộ module trên hệ thống để chắc chắn nó hoạt động ổn.
7. **Kiểm Thử Chấp Nhận**: Kiểm tra toàn bộ để đảm bảo module sẵn sàng phát hành.
8. **Báo Cáo & Sửa Lỗi**: Ghi lại các lỗi và theo dõi đến khi chúng được sửa xong.