## BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC NÔNG LÂM TP. HƠM KHOA MÔI TRƯỜNG VÀ TÀI NGUYÊN



# KHÓA LUẬN TỐT NGHIỆP

# ỨNG DỤNG GIS MÃ NGUỒN MỞ TRONG QUẢN LÝ CÂY XANH ĐÔ THỊ TẠI PHƯỜNG 6, QUẬN 3, THÀNH PHỐ HỒ CHÍ MINH

Họ và tên sinh viên: Nguyễn Quốc Tuấn Ngành: Hệ thống thông tin địa lý Niên khóa: 2007 - 2011

Tp. Hồ Chí Minh, tháng 7 năm 2011

# ỨNG DỤNG GIS MÃ NGUỒN MỞ TRONG QUẢN LÝ CÂY XANH ĐÔ THỊ TẠI PHƯỜNG 6, QUẬN 3, THÀNH PHỐ HỒ CHÍ MINH

Tác giả

#### NGUYỄN QUỐC TUẦN

Khóa luận được đề trình để đáp ứng yêu cầu cấp bằng Kỹ sư ngành

Hệ thống thông tin địa lý

Giáo viên hướng dẫn:

ThS. Quách Đồng Thắng

Trưởng Phòng Kỹ thuật – Trung tâm Ứng dụng Hệ Thống Thông Tin Địa Lý – Sở Khoa học và Công nghệ TP.HCM

TPHCM, tháng 7 năm 2011

# LỜI CẢM ƠN

Để hoàn thành luận văn này, tôi xin chân thành cảm ơn đến:

Quý thầy cô giáo Trường Đại Học Nông Lâm Tp.HCM đã tận tình dạy bảo tôi trong suốt bốn năm học.

Các thầy cô trong bộ môn Thông Tin Địa Lý Ứng Dụng đã giúp cho tôi có được những kiến thức quý báu về ngành nghề của mình cũng như những bài học kinh nghiệm từ thực tế.

Tập thể lớp Hệ thống thông tin địa lý khóa 33 đã gắn bó và giúp đỡ tôi trong suốt quá trình học cũng như trong thời gian làm luận văn tốt nghiệp đặc biệt là bạn Đặng Thị Ngọc Lý.

Các anh chị trong Trung Tâm GIS đã tận tình giúp đỡ và tạo điều kiện thuận lợi cho tôi trong suốt thời gian làm luận văn.

Gia đình và những người thân của tôi đã giúp đỡ tôi về mọi mặt để tôi có được ngày hôm nay.

Đặc biệt tôi xin chân thành cám ơn đến ThS. Quách Đồng Thắng là người đã tận tình quan tâm, giúp đỡ và hướng dẫn tôi hoàn thành luận văn này.

Xin chân thành cảm ơn

TP.HCM, tháng 7 năm 2011

Sinh viên

Nguyễn Quốc Tuấn

# TÓM TẮT LUẬN VĂN

Tên đề tài nghiên cứu: Ứng dụng GIS mã nguồn mở trong quản lý cây xanh đô thị tại phường 6, quận 3, Tp.HCM.

Khu vực nghiên cứu: Phường 6, Quận 3, Tp.HCM.

Thời gian nghiên cứu từ 15/03/2011 đến 11/07/2011.

Đối tượng nghiên cứu: Cây xanh đô thị.

Đề tài được hoàn thành bằng việc sử dụng phần mềm GIS mã nguồn mở gvSIG do gvSIG Association phát triển trên nền Java, công cụ quản lý dữ liệu không gian (shapefile) và dữ liệu thuộc tính bằng cách sử dụng DBMS mã nguồn mở hỗ trợ quản lý dữ liệu không gian như PostgreSQL/PostGIS, và môi trường lập trình eclipse trong Java.

Shapefile của các lớp cayxanh.shp, giaothong.shp, ktvhxh.shp, p6.shp, thua.shp sẽ được nhập vào hệ quản trị cơ sở dữ liệu PostgreSQL để lưu trữ. Sử dụng môi trường lập trình eclipse xây dựng chức năng quản lý cây xanh trong gvSIG cùng với các công cụ đã tích hợp sẵn trong gvSIG nhằm phục vụ cho nhu cầu quản lý.

Kết quả nghiên cứu của đề tài là một hệ thống gồm bản đồ thể hiện vị trí cây xanh và cơ sở dữ liệu thuộc tính của từng đối tượng cây xanh.

# MỤC LỤC

Trang tựa	i
Lời cảm ơn	ii
Tóm tắt luận văn	iii
Mục lục	iv
Danh sách các chữ viết tắt	v
Danh sách các hình	vi
Chương 1. MỞ ĐẦU	1
1.1. Đặt vấn đề	1
1.2. Mục đích đề tài	2
1.3. Đối tượng và phạm vi nghiên cứu	2
1.4. Ý nghĩa của đề tài	2
1.5. Nội dung và phương pháp nghiên cứu	2
1.5.1. Nội dung nghiên cứu	2
1.5.2. Phương pháp nghiên cứu	3
Chương 2. TỔNG QUAN	4
2.1. Đặc điểm khu vực nghiên cứu	4
2.2. Tổng quan các phần mềm sử dụng trong đề tài	6
2.2.1. Phần mềm GIS mã nguồn mở gvSIG	6
2.2.2. Hệ quản trị CSDL mã nguồn mở PostgreSQL	6
2.2.3. Ngôn ngữ lập trình Java và môi trường lập trình Eclipse	7
2.3. Một số công trình nghiên cứu về quản lý cây xanh đô thị ở Việt Nam	8

2.4. Một số chức năng cần có của phần mềm quản lý cây xanh	
Chương 3. KẾT QUẢ NGHIÊN CỨU	9
3.1. Tìm hiểu công cụ sẵn có trong gvSIG	9
3.1.1. Các thành phần chính của gvSIG	9
3.1.2. Công cụ chỉnh sửa trong gvSIG	9
3.1.3. Công cụ vẽ trong gvSIG	11
3.1.4. Chỉnh sửa bảng thuộc tính	
3.1.4.1. Thêm một record	13
3.1.4.2. Chỉnh sửa record	14
3.1.4.3. Xóa bỏ record	14
3.1.5. Công cụ xử lý dữ liệu	
3.2. Thiết kế CSDL	
3.2.1. Mô hình dữ liệu mức ý niệm	
3.2.2. Mô hình dữ liệu mức vật lý	
3.2.3. Ánh xạ xuống DBMS PostgreSQL/PostGIS	
3.3. Import shapefile vào PostgreSQL/PostGIS – Kết nối gvSIG với CSDL	
3.3.1. Import shapefile vào PostgreSQL/PostGIS	
3.3.2. Kết nối gvSIG với CSDL PostgreSQL/PostGIS	
3.4. Xây dựng plugin QLCX cho gvSIG trong môi trường eclipse	
3.4.1. Cấu trúc thư mục của project extQLCX	
3.4.2. Thiết kế giao diện form Báo Cáo – Thống Kê Cây Xanh	
3.4.3. Tạo cửa sổ About giới thiệu project	
3.4.3.1. Tạo trang html	
3.4.3.2. Tạo extension thể hiện about	

3.4.4. Tạo bộ cài đặt bằng IzPack	46
3.5. Chức năng của hệ thống	49
3.5.1. Công cụ cập nhật dữ liệu sử dụng công cụ sẵn có trong gvSIG	49
3.5.1.1. Công cụ thêm điểm	49
3.5.1.2. Công cụ chỉnh sửa Copying	50
3.5.1.3. Công cụ thêm đối tượng dạng đường	51
3.5.1.4. Sử dụng bảng thuộc tính chỉnh sửa đối tượng	
3.5.2. Form Báo cáo – Thống kê cây xanh	54
Chương 4. KẾT LUẬN VÀ KIẾN NGHI	55
4.1. Kết luận	55
4.2. Kiến nghị	55
TÀI LIỆU THAM KHẢO	
PHŲ LŲC	
Phụ lục 1. Hướng dẫn cài đặt công cụ QLCX	57
Phụ lục 2. Nội dụng file ThongKe.java	64

# Danh sách các chữ viết tắt

GIS: Geographic Information System – Hệ thống thông tin địa lý

CSDL: Cơ sở dữ liệu

DBMS: Database Management System – Hệ quản trị cơ sở dữ liệu

QLCX: Quản lý cây xanh

# Danh sách các hình

Hình 2.1. Bản đồ quận 3	4
Hình 2.2. Khu vực nghiên cứu phường 6, quận 3	5
Hình 3.1. Các thành phần chính của gvSIG	9
Hình 3.2. Các công cụ chỉnh sửa trong gvSIG	10
Hình 3.3. Các công cụ vẽ trong gvSIG	11
Hình 3.4. Bảng thuộc tính trong gvSIG	13
Hình 3.5. Dòng trắng xuất hiện khi thêm một record mới	13
Hình 3.6. Sửa record trong bảng thuộc tính	14
Hình 3.7. Xóa bỏ record	15
Hình 3.8. Truy cập công cụ Geoprocessing trong menu View	16
Hình 3.9. Cửa sổ Geoprocessing tools	16
Hình 3.10. Buffer zones khắp các đối tượng dạng điểm	17
Hình 3.11. Trước khi clipping lớp p6.shp	18
Hình 3.12. Kết quả sau khi clipping lớp p6.shp	19
Hình 3.13. Hai lớp p6.shp và thua.shp trước khi sử dụng công cụ Intersection	20
Hình 3.14. Kết quả sau khi sử dụng công cụ Intersection	20
Hình 3.15. Hai lớp p6.shp và thua.shp trước khi sử dụng công cụ Difference	21
Hình 3.16. Kết quả sau khi sử dụng công cụ Difference	22
Hình 3.17. Mô hình dữ liệu mức ý niệm	23
Hình 3.18. Tạo mô hình dữ liệu mức vật lý từ mô hình dữ liệu mức ý niệm	24
Hình 3.19. Chọn DBMS khi tạo mô hình dữ liệu mức vật lý	25

Hình 3.20. Mô hình dữ liệu mức vật lý	26
Hình 3.21. Tạo script file định nghĩa CSDL từ mô hình dữ liệu mức vật lý	27
Hình 3.22. Chọn đường dẫn và đặt tên cho script file	27
Hình 3.23. Chọn Query tool trong menu Tools	28
Hình 3.24. Tool bar trong pgAdmin III	28
Hình 3.25. Mở file QLCX.sql trong cửa số Query	29
Hình 3.26. Thực thi file QLCX.sql bằng lệnh Execute query	29
Hình 3.27. Cấu trúc template_postgis	30
Hình 3.28. Tạo mới GIS database trên cơ sở template_postgis	31
Hình 3.29. Menu Plugins trong PostgreSQL	31
Hình 3.30. Cửa sổ Shape File to PostGIS Importer	32
Hình 3.31. Dữ liệu bảng cây xanh trong pgAdminIII	33
Hình 3.32. Load dữ liệu không gian từ PostgreSQL/PostGIS bằng gvSIG	33
Hình 3.33. Nhập các thông số kết nối với CSDL	34
Hình 3.34. Chọn lớp dữ liệu không gian cần truy xuất	35
Hình 3.35. Tạo plugin QLCX trong eclipse	36
Hình 3.36. Chọn tham số thiết lập project mới	37
Hình 3.37. Tham chiếu các project của gvSIG	38
Hình 3.38. Chọn các project để thêm vào	39
Hình 3.39. Các project được chọn cho project QLCX	40
Hình 3.40. Cấu trúc thư mục của project extQLCX	41
Hình 3.41. Form Báo cáo – Thống kê cây xanh	42
Hình 3.42. Tạo file Thongke.java thể hiện form Báo cáo – Thống kê cây xanh	42
Hình 3.43. Cửa sổ About giới thiệu project QLCX	43

Hình 3.44. Thư mục help chứa file about giới thiệu project	44
Hình 3.45. Tạo file AboutExtension.java thể hiện about	45
Hình 3.46. Cửa sổ tiến hành cài đặt project QLCX	46
Hình 3.47. Thư mục install trong thư mục extQLCX	47
Hình 3.48. Trước khi sử dụng công cụ point để thêm điểm	49
Hình 3.49. Sau khi sử dụng công cụ point để thêm điểm với các điểm màu đỏ	50
Hình 3.50. Đối tượng dạng đường trước khi sử dụng công cụ Copying	50
Hình 3.51. Sử dụng công cụ Copying sao chép đối tượng dạng đường	51
Hình 3.52. Trước khi sử dụng công cụ Line để thêm đối tượng dạng đường	51
Hình 3.53. Sau khi sử dụng công cụ Line thêm đối tượng dạng đường	52
Hình 3.54. Dòng trắng xuất hiện khi thêm một record mới	52
Hình 3.55. Sửa record trong bảng thuộc tính	53
Hình 3.56. Xóa bỏ record	53
Hình 3.57. Form Báo cáo – Thống kê cây xanh	54

# Chương 1 MỞ ĐẦU

#### 1.1. Đặt vấn đề

Thành phố Hồ Chí Minh là thành phố có mật độ dân cư đông nhất cả nước. Cùng với sự phát triển mạnh mẽ về mặt kinh tế của thành phố là những hệ quả về mặt môi trường như ngập úng, ô nhiễm tiếng ồn, diện tích cây xanh ngày càng giảm,... do đó vấn đề quản lý cây xanh đô thị đang dần trở nên cấp thiết.

Trước đây, khi công nghệ thông tin chưa phát triển và đủ mạnh để quản lý về mặt tài nguyên thiên nhiên nói chung, vấn đề cây xanh nói riêng thì việc quản lý cây xanh đô thị bằng giấy tờ mất rất nhiều thời gian, công sức và tiền bạc. Chẳng hạn, quản lý bằng giấy tờ không giúp cho nhà quản lý có thể quản lý được thông tin của một cây xanh chặt chẽ và nhanh chóng. Tuy nhiên, Hệ thống thông tin địa lý (GIS) có thể giải quyết được những điều này. GIS ra đời vượt hẳn một số phương pháp quản lý bằng giấy tờ và đạt được những kết quả cao, là một công cụ hỗ trợ đắc lực cho nhà quản lý. Không những đáp ứng được các nhu cầu trong quản lý về thuộc tính, vị trí, tình trạng sinh trưởng, GIS còn có khả năng mô hình hóa lại thế giới thực, giúp nhà quản lý không mất nhiều thời gian và công sức trong việc giải quyết những vấn đề ngoài thực địa.

Đã có nhiều tác giả với nhiều hướng tiếp cận khác nhau trong việc ứng dụng GIS vào quản lý cây xanh đô thị. Một hướng tiếp cận ít tốn kinh phí trong việc sử dụng công nghệ và dễ dàng trong việc chia sẻ thông tin là sử dụng phần mềm GIS mã nguồn mở. Từ đó mà tôi tiến hành nghiên cứu đề tài:

### "Ứng dụng GIS mã nguồn mở trong quản lý cây xanh đô thị tại phường 6, quận 3, Tp.Hồ Chí Minh".

#### 1.2. Mục đích đề tài

 Nghiên cứu cách sử dụng phần mềm GIS mã nguồn mở gvSIG và cách phát triển ứng dụng trên gvSIG. - Áp dụng xây dựng công cụ quản lý cây xanh đô thị tại khu vực nghiên cứu.

#### 1.3. Đối tượng và phạm vi nghiên cứu

- a) Đối tượng nghiên cứu: Cây xanh đô thị.
- b) Phạm vi nghiên cứu:
- Về công nghệ: Sử dụng công nghệ mã nguồn mở cho phát triển ứng dụng desktop
   GIS trên nền hệ điều hành Microsoft Windows, gồm:
- Công cụ hiển thị, tương tác dữ liệu không gian: gvSIG.
- Hệ quản trị CSDL: PostgreSQL/PostGIS.
- Về vị trí địa lý: Khu vực nghiên cứu phát triển ứng dụng là phường 6, quận 3, thành phố Hồ Chí Minh.

#### 1.4. Ý nghĩa của đề tài

- Úng dụng được xây dựng trong đề tài sẽ là công cụ hỗ trợ đắc lực trong công tác quản lý cây xanh đô thị theo hướng tiết kiệm chi phí đầu tư xây dựng hệ thống.
- Khi ứng dụng được áp dụng trên phạm vi toàn quận và mở rộng cho toàn thành phố thì sẽ hỗ trợ công tác quản lý quy hoạch cây xanh đô thị trên địa bàn toàn thành phố Hồ Chí Minh.

#### 1.5. Nội dung và phương pháp nghiên cứu

#### 1.5.1. Nội dung nghiên cứu

a) Nghiên cứu cách sử dụng phần mềm GIS mã nguồn mở gvSIG và cách phát triển ứng dụng trên gvSIG.

Mô tả:

 Nghiên cứu các công cụ sẵn có trong phần mềm gvSIG và cách kết nối gvSIG với cơ sở dữ liệu.

- Tìm hiểu hệ quản trị cơ sở dữ liệu mã nguồn mở PostgreSQL và phần mở rộng PostGIS.
- Tìm hiểu ngôn ngữ lập trình Java và môi trường lập trình Eclipse.
- Cách xây dựng một plugin cho gvSIG trên môi trường Eclipse.
- b) Thiết kế CSDL và xây dựng công cụ phục vụ cho đề tài.

Mô tả:

- Phân tích và thiết kế cơ sở dữ liệu từ mức ý niệm đến mức vật lý.
- Thiết kế giao diện.
- Tích hợp dữ liệu không gian và dữ liệu thuộc tính vào hệ thống.
- Cài đặt và chạy thử nghiệm ứng dụng.

#### 1.5.2. Phương pháp nghiên cứu

Các phương pháp sử dụng trong đề tài như sau:

- Thu thập và tổng hợp tài liệu (nghiên cứu về gvSIG phục vụ cho việc phát triển ứng dụng).
- Thu thập dữ liệu: Dữ liệu được thu thập từ Phòng Kỹ Thuật, Trung Tâm Ứng Dụng GIS thuộc Sở Khoa học và Công nghệ Tp.HCM.
- Thiết kế CSDL theo mô hình thực thể kết hợp.

## Chương 2

### TỔNG QUAN

#### 2.1. Đặc điểm khu vực nghiên cứu

Quận 3 với vị trí là một quận nằm ở trung tâm thành phố Hồ Chí Minh với diện tích khoảng 4,92 km<sup>2</sup>, dân số: 189.764 người (theo điều tra ngày 1/4/2009). Quận 3 với địa giới hành chánh: phía Bắc giáp quận Phú Nhuận và quận Tân Bình, phía Đông và phía Nam giáp quận 1, phía Tây giáp quận 10, được chia làm 14 phường, đánh số từ 1 đến 14.



Hình 2.1. Bản đồ quận 3

Về giao thông đường bộ, mật độ đường sá dày đặc, có nhiều trục đường giao thông quan trọng chạy ngang qua như : đường Cách Mạng Tháng 8 nối với Quốc lộ 22 đi Tây Ninh sang Campuchia, đường Điện Biên Phủ nối với Quốc lộ 1 xuyên Việt, đường Nam Kỳ Khởi Nghĩa đi sân bay Tân Sơn Nhất.



Hình 2.2. Khu vực nghiên cứu phường 6, quận 3

Khu vực nghiên cứu là phường 6, quận 3 là phường có mật độ cây xanh tương đối cao và dày đặc. Phường 6 gồm có một số con đường lớn như: Nguyễn Thị Minh Khai, Võ Văn Tần, Nguyễn Đình Chiểu, Pastuer, Phạm Ngọc Thạch...

Tính đến hết thời điểm 31/12/2005 thì phường 6 có: 2152 hộ, 9396 người, 4 khu phố, 55 tổ dân phố, mật độ dân số là 10639 người/km<sup>2</sup> (theo số liệu của <u>http://www.quan3.hochiminhcity.gov.vn/default.aspx</u>). Và theo dữ liệu thu thập được thì phường 6 có khoảng 1492 cá thể cây xanh do Khu quản lý giao thông số 1 quản lý.

#### 2.2. Tổng quan các phần mềm sử dụng trong đề tài

#### 2.2.1. Phần mềm GIS mã nguồn mở gvSIG

Các phần mềm mã nguồn mở đã thực sự trưởng thành trong những năm qua và đã trở thành những công cụ thiết thực và hỗ trợ tốt cho người dùng. Các phần mềm mã nguồn

mở cho GIS cũng không nằm ngoài xu hướng đó và đang được sử dụng ngày một rộng rãi bên cạnh các phần mềm thương mại.

Một trong những phần mềm GIS mã nguồn mở đang được cộng đồng GIS ưa dùng là gvSIG. gvSIG có giao diện thân thiện với người dùng, hỗ trợ quản lý dữ liệu vector và raster, một hệ thống phát triển theo dạng từng modul riêng lẻ và hiện nay gvSIG đã có thể đáp ứng nhu cầu cho nhiều đối tượng người dùng khác nhau.

#### 2.2.2. Hệ quản trị CSDL mã nguồn mở PostgreSQL

PostgreSQL là một hệ quản trị CSDL quan hệ - đối tượng có nguồn gốc từ dự án Ingres tại trường Đại Học Berkeley của Đức vào năm 1986. Với PostGIS - thành phần mở rộng của PostgreSQL giúp cho PostgreSQL có khả năng lưu trữ các dữ liệu địa lý như điểm, đường, vùng,...

#### Công cụ quản trị PostgreSQL:

a/ Công cụ dòng lệnh psql:

Công cụ dòng lệnh psql là công cụ dùng để tạo và quản lý bằng dòng lệnh. Công cụ này cho phép người dùng:

- Kết nối tới cơ sở dữ liệu.
- Thi hành truy vấn.
- Quản lý cơ sở dữ liệu: tạo cơ sở dữ liệu, thêm bảng, truy cập hay cập nhật dữ liệu sử dụng các lệnh SQL.

b/ Công cụ đồ họa pgAdmin III

pgAdmin III là một giao diện đồ họa cho cơ sở dữ liệu trong PostgreSQL. Đây là công cụ quản lý và phát triển cơ sở dữ liệu PostgreSQL mạnh mẽ, miễn phí và cung cấp nhiều tính năng:

– Tạo và xóa tablespaces, database, tabales và schemas.

- Thi hành lệnh SQL với cửa sổ truy vấn.
- Xuất kết quả truy vấn SQL thành các tập tin.
- Sao chép, phục hồi database hoặc tables.
- Xem, biên tập và thêm dữ liệu vào table.

PostGIS là phần mở rộng của PostgreSQL dùng để quản lý dữ liệu không gian. PostGIS hỗ trợ các phép truy vấn và phân tích không gian hoàn toàn bằng dòng lệnh SQL.

#### 2.2.3. Ngôn ngữ lập trình Java và môi trường lập trình Eclipse

Java là ngôn ngữ lập trình do James Gosling của Sun Microsystem đưa ra. Java là một ngôn ngữ lập trình dạng lập trình hướng đối tượng (OOP). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Eclipse là một môi trường phát triển tích hợp (Integrated Development Environment – IDE) cho Java, được phát triển ban đầu bởi IBM và hiện nay bởi tổ chức **Eclipse** (<u>http://www.eclipse.org</u>). Eclipse có thể tích hợp với nhiều công cụ hỗ trợ khác để có được một bộ công cụ hòan chỉnh mà không cần dùng đến phần mềm riêng nào khác.

#### 2.3. Một số công trình nghiên cứu về quản lý cây xanh đô thị ở Việt Nam

Ở nước ta hiện đã có khá nhiều công trình nghiên cứu bảo vệ và quản lý hệ thống cây xanh, tiêu biểu như:

- Chương trình quản lý cây xanh trên các đường phố thủ đô Hà Nội, Tp.HCM, Nha Trang,...
- Phần mềm quản lý cây xanh do Công ty cây xanh (thuộc Sở Giao thông công chính Tp.Đà Nẵng) phối hợp với Trung tâm Công nghệ phần mềm thành phố xây dựng và triển khai thực hiện từ cuối tháng 9/2006.

- Tổ chức Cộng đồng Châu Âu tài trợ Công ty công trình đô thị Trà Vinh thực hiện dự án bảo vệ cây xanh và trồng mới 20.000 cây xanh trên địa bàn thị xã Trà Vinh.
- Chương trình quản lý cây xanh trong khu Đại Nội của cung đình Huế.

#### 2.4. Một số chức năng cần có của phần mềm quản lý cây xanh

Phần mềm quản lý cây xanh đô thị theo Wagar và Smiley (1990) cần có các chức năng

- 1. Truy xuất, trình bày và kiểm tra dữ liệu.
- Thiết lập thứ tự công việc như: sắp xếp lịch bảo dưỡng cây xanh, công nhân bảo dưỡng theo tuyến đường,...
- 3. Quản lý thông tin về cây xanh như: chiều cao của cây, tán cây, tuổi cây,...
- 4. Tổng kết thông tin.
- 5. Bản đồ vị trí cây xanh.
- 6. Tạo đồ thị.
- 7. Theo dõi chi phí bảo quản và tình trạng sinh trưởng của những loài cây khác nhau.
- 8. Dự báo khối lượng công việc trong tương lai.

## Churong 3

## KẾT QUẢ NGHIÊN CỨU

#### 3.1. Tìm hiểu công cụ sẵn có trong gvSIG

#### 3.1.1. Các thành phần chính của gvSIG

Mỗi phiên làm việc, gvSIG sẽ mở một project hoặc tạo một project mới. Mỗi project trong gvSIG chứa các thông tin: các khung nhìn (view), các bảng dữ liệu (table), các trang in bản đồ (map).

Mỗi dự án có thể có nhiều khung nhìn khác nhau, trong đó các khung nhìn khác nhau có thể có các lớp bản đồ khác nhau.



Hình 3.1: Các thành phần chính của gvSIG

- View: là nơi thể hiện các lớp bản đồ và thực hiện các phép toán phân tích không gian.
- Tables: là nơi chứa các dữ liệu thuộc tính.
- Maps: là nơi trình bày trang in bản đồ.

#### 3.1.2. Công cụ chỉnh sửa trong gvSIG

Công cụ chỉnh sửa là một tập hợp các công cụ dùng để chỉnh sửa hoặc thay đổi các yếu tố không gian (điểm, đường, vùng).

gvSIG sử dụng 3 cách khác nhau để chạy những công cụ này:

- Click vào nút tương ứng trên thanh tool.
- Từ thanh menu.
- Viết các lệnh trong giao diện điều khiển lệnh.

#### Một số công cụ chỉnh sửa:

Công cụ được truy cập từ menu Geometry  $\rightarrow$  Modify



Hình 3.2: Các công cụ chỉnh sửa trong gvSIG

- Công cụ Selecting: Gồm 2 loại là Simple selection và Complex selection.
- Công cụ Copying <sup>1</sup> : Tạo ra một bản sao của đối tượng được chọn. Bản sao sẽ cùng hướng và kích thước với bản gốc.
- Công cụ Symmetry<sup>44</sup>: Tạo ra một đối tượng đối xứng với đối tượng được chọn.
- Công cụ Rotating <sup>1</sup>: Xoay các đối tượng được chọn qua một điểm làm tâm.
- Công cụ Scaling <sup>1</sup>: Thay đổi kích thước của đối tượng được chọn.
- Công cụ Moving Elements <sup>1</sup> : Di chuyển đối tượng.
- Công cụ Editing Vertex <sup>3</sup>: Cho phép duyệt qua các đỉnh của đối tượng thực hiện các thao tác như thêm hay xóa đỉnh.
- Công cụ Internal Polygon 2: Cho phép tạo ra một đa giác nằm trong một đối tượng sẵn có.

#### 3.1.3. Công cụ vẽ trong gvSIG

gvSIG gồm các đối tượng vẽ cơ bản như đường, đường tròn, đa giác,...

Lớp cần chỉnh sửa có đối tượng khác nhau thì sẽ sử dụng công cụ chèn đối tượng thích hợp với đối tượng đó. Ví dụ như lớp là đối tượng điểm chỉ được chèn thêm điểm chứ không chèn thêm bất kỳ đối tượng nào khác.

Một số công cụ vẽ:



Công cụ được truy cập từ menu Geometry → Insert

Hình 3.3: Các công cụ vẽ trong gvSIG

- Công cụ Point : Để thêm điểm thì lớp đang chỉnh sửa phải là lớp đối tượng dạng điểm.
- Công cụ Multipoint <sup>1</sup> Tạo ra một tập điểm có cùng chức năng với nhau như là một đối tượng đơn, tức là một điểm được chọn thì các điểm còn lại đều được chọn.
- Công cụ Arc <sup>C</sup>: Tạo ra một cung với 3 điểm.

- Công cụ Polyline <sup>2</sup>: Tạo ra một tập hợp nhiều cung và (hoặc) nhiều đoạn thẳng kết hợp với nhau.
- Công cụ Polygon <sup>O</sup>: Tạo ra một đa giác. gvSIG cho phép xác định số cạnh của đa giác cần tạo.
- Công cụ Rectangle <sup>□</sup>: Tạo ra một hình chữ nhật.
- Công cụ Square <sup>□</sup>: Tạo ra một hình vuông.
- Công cụ Circle <sup>O</sup>: Tạo ra một đường tròn.
- Công cụ Ellipse <sup>O</sup>: Tạo ra một ellipse.

#### 3.1.4. Chỉnh sửa bảng thuộc tính

Bảng là một phần của cơ sở dữ liệu. Bảng gồm các hàng hay các record và các cột hay các trường chứa thông tin thuộc tính của các đối tượng (điểm, đường, vùng).

🐗 Table: Attribute table: VNM_Provin 🔳 🗖 🔀					
Country	ADMName	ADMUnitID	ADMParent	][	
Vietnam	Ha Nam	70400002	70400001	1 🔨	
Vietnam	Nam Dinh	70400123	70400001	1	
Vietnam	Thai Binh	70400360	70400001	1	
Vietnam	Ninh Binh	70400654	70400001	1	
Vietnam	Ha Noi	70400805	70400001	1	
Vietnam	Hai Phong	70401046	70400001	1	
Vietnam	Ha Tay	70401277	70400001	1	
Vietnam	Hai Duong	70401616	70400001	1	
Vietnam	Hung Yen	70401892	70400001	1	
Vietnam	Tuyen Quang	70402064	70402063	1	
Vietnam	Ven Bai	70402216	70402063		
				>	
0 / 61 Total red	ords selected.				

Hình 3.4: Bảng thuộc tính trong gvSIG

Mỗi yếu tố (điểm, đường, vùng) của một lớp tương ứng một record trong bảng thuộc tính.

#### 3.1.4.1. Thêm một record

Để thêm một record mới, một dòng trắng sẽ xuất hiện trong bảng thuộc tính.

	🐝 Table: Attribute table: VNM				<
	Country	ADMName	ADMUnitID	ADMPar	
	Vietnam	Soc Trang	70410124	70409614	^
	Vietnam	Long An	70410230	70409614	
	Vietnam	Bac Lieu	70410428	70409614	
	Vietnam	Ca Mau	70410485	70409614	
	Vietnam	Dong Thap	70410569	70409614	
A	Vietnam	An Giang	70410720	70409614	
<u>i</u> <u>4</u> <u></u> <	Vietnam	Tien Giang	70410872	70409614	
	Vietnam	Vinh Long	70411045	70409614	
				1	~
	<			>	
	1 / 62 Total rec	ords selected.			

Hình 3.5: Dòng trắng xuất hiện khi thêm một record mới

#### 3.1.4.2. Chỉnh sửa record

Để sửa đổi record của một đối tượng, chọn đối tượng và đi đến bảng thuộc tính, khi đó record sẽ sáng lên tương ứng với đối tượng được chọn. Kích chọn cột cần chỉnh sửa trong record.

🐝 Table: /	Attribute tab	le: VNM		
Country	ADMName	ADMUnitID	ADMPar	4
Vietnam	Soc Trang	70410124	70409614	
Vietnam	Long An	70410230	70409614	
Vietnam	Bac Lieu	70410428	70409614	
Vietnam	Ca Mau	70410485	70409614	
Vietnam	Dong Thap	70410569	70409614	
Vietnam	An Giang	70410720	70409614	
Vietnam	Tien Giang	70410872	70409614	
Vietnam	Vinh Long	70411045	70409614	
			~	
<		•	>	
1 / 62 Total re	ecords selected.			

Hình 3.6: Sửa record trong bảng thuộc tính

#### 3.1.4.3. Xóa bỏ record

Để xóa bỏ record của một đối tượng, chọn đối tượng và đi đến bảng thuộc tính (hay chọn trực tiếp trong bảng thuộc tính), đến menu Table  $\rightarrow$  Remove row.



Hình 3.7: Xóa bỏ record

#### 3.1.5. Công cụ xử lý dữ liệu

Phần mở rộng xử lý dữ liệu trong gvSIG giúp cho gvSIG tạo ra các lớp thông tin vector mới.

Để truy cập các công cụ này, chọn nút "Geoprocessing tools" 🐼. Hoặc vào menu View

 $\rightarrow$  Geoprocessing tools



Hình 3.8: Truy cập công cụ Geoprocessing trong menu View

Một cửa sổ sẽ xuất hiện.



Hình 3.9: Cửa số Geoprocessing tools

Khi một công cụ xử lý được chọn thì một đoạn mô tả về công cụ đó sẽ xuất hiện bên phải và nút "open tool..." sáng lên.

Một số công cụ xử lý dữ liệu không gian:

1/ Buffer zones:

Công cụ này tạo ra "vùng ảnh hưởng" xung quanh đối tượng vector (điểm, đường, vùng) của lớp nhập vào.

Trong trường hợp đối tượng nhập vào là đa giác, vùng ảnh hưởng có thể nằm bên ngoài đa giác, bên trong đa giác hay cả bên trong và bên ngoài.



Hình 3.10: Buffer zones khắp các đối tượng dạng điểm

2/ Clipping:

Công cụ này cho phép giới hạn vùng làm việc của một lớp vector (điểm, đường, vùng) và trích ra vùng quan tâm.

Để thực thi công cụ này thì cần một lớp đưa vào (lớp sử dụng để trích vùng ra) và một lớp cắt (clipping layer) xác định vùng làm việc.

Tiến trình kiểm tra tất cả các đối tượng vector trong lớp đưa vào và sẽ tính toán giao nhau của các đối tượng vector trong vùng làm việc xác định bởi lớp cắt để lớp kết quả chỉ cho ra các đối tượng trong vùng làm việc.



Ví dụ: Lớp đưa vào là p6.shp, sử dụng lớp thua.shp làm lớp cắt.

Hình 3.11: Trước khi clipping lớp p6.shp



Hình 3.12: Kết quả sau khi clipping lớp p6.shp

3/ Intersection:

Công cụ này phải thực hiện trên 2 lớp: lớp nhập vào (Input layer) và lớp chồng xếp (overlay layer), và các đối tượng của 2 lớp phải ở dạng polygon.

Ví dụ: Sử dụng công cụ intersection cho 2 lớp p6.shp và lớp thua.shp



Hình 3.13: Hai lớp p6.shp và thua.shp trước khi sử dụng công cụ Intersection



Hình 3.14: Kết quả sau khi sử dụng công cụ Intersection

4/ Difference:

Công cụ này làm việc trong 2 lớp: lớp nhập vào và lớp chồng xếp. Các đối tượng trong cả 2 lớp phải ở dạng polygon chứa những vùng trong một lớp mà không có ở lớp khác.



Ví dụ: Sử dụng công cụ Difference cho 2 lớp p6.shp và thua.shp

Hình 3.15: Hai lớp p6.shp và thua.shp trước khi sử dụng công cụ Difference



#### 3.2. Thiết kế CSDL

Đề tài tiến hành phân tích, thiết kế CSDL với sự hỗ trợ của phần mềm PowerDesigner, gồm các nội dung:

- Thiết kế mô hình dữ liệu mức ý niệm conceptual data model (\*.cdm).
- Tạo mô hình dữ liệu mức vật lý physical data model (\*.pdm).
- Ánh xạ mô hình dữ liệu mức vật lý xuống DBMS PostgreSQL/PostGIS.

3.2.1. Mô hình dữ liệu mức ý niệm



Hình 3.17: Mô hình dữ liệu mức ý niệm

CSDL được thiết kế ở mức ý niệm với 5 thực thể cần quan tâm là: CAYXANH, BAODUONG, KTVHXH, THUA, GIAOTHONG. Trong đó, CAYXANH và BAODUONG có mối quan hệ với nhau.

Mô tả mối quan hệ CAYXANH và BAODUONG



Mô tả: Một CAYXANH (cây xanh) có một hay nhiều lần được bảo dưỡng. Mỗi lần BAODUONG (bảo dưỡng) sẽ bảo dưỡng một hoặc nhiều cây xanh. Mối quan hệ giữa loại thực thể CAYXANH và BAODUONG là mối quan hệ \* - \* (nhiều – nhiều).

Sau khi có mô hình dữ liệu mức ý niệm, công cụ Generate Physical Data Model của phần mềm PowerDesigner được sử dụng để tạo mô hình dữ liệu mức vật lý.

Vào menu Tools  $\rightarrow$  chọn Generate Physical Data Model

📲 File Edit View Model Symbol Database	e Report Repository	Tools Window Help	
🖹 🗃 💕 🖌 🥥 🖪 🗛 🕹 🕹 🕿	5 × 9 @ m	Complete Links	Ctrl+F5
Workspace		Check Model Impact and Lineage Analysis Compare Models	F4 Ctrl+F11 Ctrl+F6
PDM_QLCX *	Mas	Merge Model Extended Generation	Shift+F6
	Ten. Nam	Generation Links	•
	Phar Ghio	Spell Checking Options	
		Execute Commands	•
		Mapping Editor	
		XML Builder Wizard	Ctrl+Shift+W
		Generate Physical Data Model	Ctrl+Shift+P
		Generate Conceptual Data Model	Ctrl+Shift+C
		Generate Logical Data Model	Ctrl+Shift+L
		Generate Object-Oriented Model	Ctrl+Shift+O
	bd a	Generate XML Model	Ctrl+Shift+M

Hình 3.18: Tạo mô hình dữ liệu mức vật lý từ mô hình dữ liệu mức ý niệm

Tiếp theo là chọn DBMS là PostgreSQL, phần mềm sẽ tự động chuyển mô hình dữ liệu mức ý niệm sang mô hình dữ liệu mức vật lý.

PDM	Generatio	n Options	
Ger	neral DBMS	Preserve Options Detail Target Models Selection	
	💿 Generate n	ew Physical Data Model	
1	DBMS:	PostgreSQL 8 🔽 🔽	
		<ul> <li>Share the DBMS definition</li> </ul>	
		Copy the DBMS definition in model	
l I	Name:	PDM_QLCX2 =	
(	Code:	PDM_QLCX2	
		Configure Model Options	
	🔿 Update exi	sting Physical Data Model	
9	Select model:	<none></none>	
1	DBMS:		
		✓ Preserve modifications	
	•	OK Cancel Apply	Help

Hình 3.19: Chọn DBMS khi tạo mô hình dữ liệu mức vật lý

#### 3.2.2. Mô hình dữ liệu mức vật lý



Hình 3.20: Mô hình dữ liệu mức vật lý

Do hai thực thể CAYXANH và BAODUONG có mối quan hệ nhiều – nhiều nên khi chuyển sang mô hình dữ liệu mức vật lý từ mô hình dữ liệu mức ý niệm sẽ tạo ra một thực thể của mối quan hệ này, đặt tên thực thể là BAODUONGCAY (bảo dưỡng cây).

#### 3.2.3. Ánh xạ xuống DBMS PostgreSQL/PostGIS

Từ mô hình dữ liệu mức vật lý, công cụ Generate Database của phần mềm PowerDesigner được sử dụng để tạo script file sẵn dùng cho việc ánh xạ mô hình dữ liệu mức vật lý xuống hệ quản trị CSDL PostgreSQL/PostGIS.

Vào menu Database  $\rightarrow$  chọn Generate Database
•1 PowerDesigner - [PDM PDM_C	LCX , Dia	gram_1	- C:\Docu	ments	and Set	tings\Ql
📲 File Edit View Model Symbol	Database	Report	Repository	Tools	Window	Help
	Change Edit Curi Default I	Current [ rent DBM: Physical (	DBMS S Options			٥
Workspace	Generat	e Databa	se		Ctrl+G	
ingram_1 ⊕ ingram_1	Apply M Update I	odel Char Model fro	nges to Datab m Database.	)ase	Ctrl+R	4
	Estimate Database Size Generate Test Data Generate Extraction Scripts				Ctrl+Shift-	+D 00
Tables	Execute	SQL			Ctrl+Shift	+E
⊞ ि References	Configur Connect Disconne Connect	re Connec  ect ion Infori	rtions		Ctrl+Shift	LC. +N G
						ler

Hình 3.21: Tạo script file định nghĩa CSDL từ mô hình dữ liệu mức vật lý

Tiếp theo, chọn đường dẫn và đặt tên script file (QLCX.sql) → chọn OK.

Database Gener	Database Generation - PDM_QLCX (PDM_QLCX)					
General Options	Format Selection Summary Preview					
DBMS:	Sybase SQL Anywhere 1	ybase SQL Anywhere 11				
Directory:	E:\Quản lý cây xanh\		▼			
File name:	QLCX.sq		►			
Generation type:	<ul> <li>Script generation</li> </ul>	Cone file only				
	O Direct generation					
	Check model	Automatic archive				
- Quick launch -						
Selection:	<default></default>		~			
Settings set:	Image: Contract of the second sec					
-		OK Cancel Apply	Help			

Hình 3.22: Chọn đường dẫn và đặt tên cho script file

Tạo database: Khởi động PostgreSQL, chọn CSDL "qlcx", vào menu Tools → Query tool



Hình 3.23: Chọn Query tool trong menu Tools

Hoặc chọn Execute arbitrary SQL queries trên Tool bar.

🔌 🛃 🔯	Execute arbitrary SQL queries.
-------	--------------------------------

Hinh 3.24: Tool bar trong pgAdmin III

Tiếp theo, cửa sổ Query xuất hiện, vào menu File → Open, chọn đường dẫn đến file QLCX.sql



Hình 3.25: Mở file QLCX.sql trong cửa sổ Query

Tiếp theo, thực thi truy vấn bằng cách chọn Execute Query trên tool bar



Hình 3.26: Thực thi file QLCX.sql bằng lệnh Execute query

Sau khi thực thi lệnh Execute query, các thực thể sẽ được cài đặt trên hệ quản trị CSDL PostgreSQL/PostGIS.

#### 3.3. Import shapefile vào PostgreSQL/PostGIS – Kết nối gvSIG với CSDL

#### 3.3.1. Import shapefile vào PostgreSQL/PostGIS

Khi cài đặt PostgreSQL, lưu ý chọn các Extension: PostGIS (hỗ trợ dữ liệu không gian), PgAdmin III (giao diện đồ họa quản trị PostgreSQL), psql (giao diện command line). Sau khi cài đặt, extension PostGIS sẽ tạo ra một PostGIS database template với tên là template\_postgis, với các table, function,... sẵn sàng hỗ trợ quản lý dữ liệu không gian.



Hình 3.27: Cấu trúc template\_postgis

Login vào PostgreSQL bằng pdAdmin III, chọn "New database" để tạo một database hoàn toàn mới, hỗ trợ dữ liệu không gian, đặt tên là qlcx. Mục Owner chọn mặc định là Postgres, mục Template chọn template\_postgis.

Object browser		× Properties Statistics Depe			
Servers (1)	🚺 New Database				
	Properties Variable Name OID	es Privileges SQL			
	Owner	postgres			
<ul> <li>allespaces (2)</li> <li>allespace (2)<!--</td--><td>Template</td><td>template_postgis</td></li></ul>	Template	template_postgis			
	Tablespace Schema restriction	<pre><default tablespace=""></default></pre>			
	Collation Character type	<ul> <li>✓</li> <li>✓</li> </ul>			
	Connection Limit	-1			
	Comment				
	Help	OK Cancel			

Hình 3.28: Tạo mới GIS database trên cơ sở template\_postgis

Chọn CSDL "qlcx" vừa mới tạo, vào menu Plugins chọn PostGIS Shapefile and DBF loader để Import shapefile vào CSDL "qlcx".



Hinh 3.29: Menu Plugins trong PostgreSQL

Xuất hiện cửa sổ Shape File to PostGIS Importer, chọn shapefile và Import vào.

🗳 Shape File to PostGIS Importer					
Shape File					
Cayxanh.shp					
PostGIS Connection					
Username:	postgres				
Password:	•••••				
Server Host:	localhost		5432		
Database:	thinghiem				
	Test Con	nection			
Configuration					
Destination Schema: public		SRID:		-1	
Destination Table: cayxa	nh	Geometry Co	lumn:	the_geo	om
Ontions	moort	About			Cancel
		About			
					]
Import Log					
					$\sim$

Hình 3.30: Cửa sổ Shape File to PostGIS Importer

Kết quả là table "cayxanh" và import đầy đủ các records của shapefile cayxanh.shp

🔲 Edit	📕 Edit Data - PostgreSQL 8.4 (localhost:5432) - qlcx - cayxanh_shp 📃 🗖 🔀							
File Edit View Tools Help								
: 🔳   i	🗄 🔳   🕮 🦚   🛍   🖷   🐨   🎖 🕴 No limit 🛛 💌							
	gid integer	id [PK] integer	ma_so_cay character var	don_vi_ql character var	so_nha character var	ten_doan_d character var	ten_dı charac	^
1	14	182	244A	Khu Quản lý gia:	196A	Ngô Thời Nhiệm	Nam Kỳ	
2	15	183	244	Khu Quản lý gia:	196A	Ngô Thời Nhiệm	Nam Kỳ	
3	16	184	242	Khu Quản lý gia:	194	Nguyễn Đình Ch	Nam Kỳ	
4	17	185	295	Khu Quản lý gia:	155T	Nguyễn Đình Ch	Nam Kỳ	
5	18	186	236	Khu Quản lý gia:	192	Nguyễn Đình Ch	Nam Kỳ	
6	19	187	234A	Khu Quản lý gia:	192	Nguyễn Đình Ch	Nam Kỳ	
7	20	193	254	Khu Quản lý gia:	208	Ngô Thời Nhiệm	Nam Kỳ	
8	21	194	256	Khu Quản lý gia:	218	Ngô Thời Nhiệm	Nam Kỳ 🛛	~
<							>	
Scratch p	ad						;	×
							1	
1								2
1493 row	s.							

Hình 3.31: Dữ liệu bảng cây xanh trong pgAdmin III

#### 3.3.2. Kết nối gvSIG với CSDL PostgreSQL/PostGIS

Gồm những bước sau:

Bước 1: Khởi động gvSIG → vào View → Add layer

	Window	Help	View	Layer	
ĺ			<b>4</b>	Add layer Alt+O	
			Properties		
			Export		•

Hình 3.32: Load dữ liệu không gian từ PostgreSQL/PostGIS bằng gvSIG

- − Bước 2: Chọn thẻ GeoDB → kích chọn Add connection  $\boxed{\square}$
- Bước 3: Chọn các thông số kết nối với CSDL

С	Connection parameters				
	Connection settings	]			
	Connection name:	qlcx			
	Driver:	PostGIS JDBC Driver			
	Server URL:	localhost			
	Port:	5432			
	DB name:	qlcx			
	Warning: you must input the exact name (this difference between capital letters and small letters).				
	User:	postgres			
	Password:	•••••			
	Connected:				
	Ok	Cancel			

Hình 3.33: Nhập các thông số kết nối với CSDL

- Bước 4: Chọn lớp dữ liệu không gian cần truy xuất

Add layer
File GeoDB
Choose connection         [C] qlcx (PostGIS JDBC Driver)         Choose table         public.baoduongcay         public.cayxanh         public.cayxanh         public.cayxanh_shp         public.geography_columns         public.geometry_columns         public.geometry_columns         public.giaothong         v ten_duong [varchar]         v ten_duong [varchar]         v ten_bhuong [varchar]         v public.ktvhxh         v public.ktvhxh         v public.p6_shp
SQL restriction
Working area Get view
Maximum Y Minimum X Minimum X
Ok Cancel

Hình 3.34: Chọn lớp dữ liệu không gian cần truy xuất

## 3.4. Xây dựng plugin QLCX cho gvSIG trong môi trường eclipse

Mở workspace của gvSIG trong Eclipse, chọn menu File  $\rightarrow$  New  $\rightarrow$  Java Project

Đặt tên project, các thông tin còn lại để mặc định

🖨 New Java Project			
Create a Java Project Create a Java project in the workspace or in an external location.			
Project name: exQLCX  Use default location	Browse		
JRE • Use an execution environment JRE:	JavaSE-1.6		
Use a project specific JRE:	jre6		
Project layout     OUse project folder as root for sources     OCreate separate folders for sources a	; and class files and class files <u>Configure default</u>		
Working sets Add project to working sets Working sets:	Select		
? < Back	Next > Finish Cancel		

Hinh 3.35: Tạo plugin QLCX trong eclipse

Nhấn nút Next > để tiếp tục



Hình 3.36: Chọn tham số thiết lập project mới

Trong tab <sup>(#)</sup> <sup>Source</sup> nên để các thông tin mặc định, không cần chỉnh sửa. Chuyển sang tab <sup>(=)</sup> <sup>Projects</sup> để tham chiếu các project cần thiết.

🖨 New Java Project	
<b>Java Settings</b> Define the Java build settings.	
😕 Source 🗁 Projects 🛋 Libraries 🍫	Order and Export
	Add Edit Remove
Rext >	Finish Cancel

Hình 3.37: Tham chiếu các project của gvSIG

Trong tab Projects , nhấn nút <u>Add...</u> để Eclipse hiển thị danh sách các project trong workspace.

🖨 Required Project Select	ion 📃 🗖 🔀
Select projects to add:	
	Select All Deselect All
	OK Cancel

Hình 3.38: Chọn các project để thêm vào

Chọn các project sau:

- \_fwAndami
- appgvSIG
- libCorePlugin
- libExceptions
- libFMap

Kết quả như hình sau:

🖨 New Java Project	
Java Settings Define the Java build settings.	
<ul> <li>Source Projects Libraries</li> <li>Required projects on the build path:</li> <li>         _fwAndami     </li> <li>         _appgvSIG     <li>         _ibCorePlugin     <li>         _ibExceptions     </li> <li>         _ibFMap     </li> </li></li></ul>	Corder and Export Add Edit Remove
Rext >	Finish Cancel

Hình 3.39: Các project được chọn cho project QLCX

Sau đó nhấn nút <u>Einish</u> để hoàn tất quá trình tạo project mới.

## 3.4.1. Cấu trúc thư mục của project extQLCX



### Hình 3.40: Cấu trúc thư mục của project extQLCX

Thư mục extQLCX gồm một số thư mục con sau:

- Thư mục config: Gồm có hai file: config.xml và connect.txt
- Thư mục help: Gồm có hai file: about.htm và about.jpg
- Thư mục install: Gồm 2 file build.xml và install.xml và thư mục con là resources.
- Thư mục lib: Chứa các file dùng tham chiếu thư viện JasperReports dùng để lập báo cáo.
- Thư mục src: Chứa các file .java dùng để thể hiện giao diện của form báo cáo và kết nối với CSDL.
- Thư mục templates: Chứa file qlcx.jasper và qlcx.jrxml là file thể hiện biểu mẫu báo cáo của project.
- File build.xml khai báo các thông số, dòng lệnh biên dịch project extQLCX.

#### 3.4.2. Thiết kế giao diện form Báo Cáo – Thống Kê Cây Xanh

hống Kê Cây Xanh							_	
Xanh								
Dầu 🗸	Danh S	Sách T	hống Ké	ê				
Nam Kỳ Khởi Nghĩa 🛛 🗸	STT	Sõ	Tê Nam	Tê Dầu	Nă 2008	Lề Chẳn	Gh	
2008	2	186	Nam	Dầu	2008	Chẵn		
Chẵn 🗸	3	196A 196A	Nam Nam	Dau Dầu	2008 2008	Chan Chẵn		
	5	208	Nam	Dầu	2008	Chẵn		
Xuất Báo Cáo	7	218 208	Nam Nam	Dau Dầu	2008	Chân Chẵn		
	hống Kê Cây Xanh Xanh Dầu v Nam Kỳ Khởi Nghĩa v 2008 v Chẵn v Xuất Báo Cáo	hống Kê Cây Xanh Xanh Dầu v Nam Kỳ Khởi Nghĩa v 2008 v Chẵn v Xuất Báo Cáo 7	hống Kê Cây Xanh Xanh Dầu v Nam Kỳ Khời Nghĩa v 2008 v Chẵn v Xuất Báo Cáo	hống Kê Cây Xanh Xanh Dầu ✔ Nam Kỳ Khởi Nghĩa ✔ 2008 ✔ Chẵn ✔ Xuất Báo Cáo	hống Kê Cây Xanh Xanh Dầu Nam Kỳ Khởi Nghĩa 2008 Chẵn Xuất Báo Cáo Nam Li 190 Nam Dầu 2 186 Nam Dầu 2 186 Nam Dầu 3 196A Nam Dầu 5 208 Nam Dầu 6 218 Nam Dầu 7 208 Nam Dầu	bằng Kê Cây Xanh         Xanh         Dầu       Vanh Sách Thống Kê         Nam Kỳ Khởi Nghĩa       STT       Sã       Tê       Tê       Nă         2008       V       1       190       Nam       Dầu       2008         Chẵn       V       1       190       Nam       Dầu       2008         Chẵn       V       1       190       Nam       Dầu       2008         Xuất Báo Cáo       Xuất Báo Cáo       Xuất Báo Cáo       Nam       Dầu       2008	hống Kê Cây Xanh Xanh Dầu ✔ Nam Kỳ Khời Nghĩa ✔ 2008 ✔ Chẵn ✔ Xuất Báo Cáo	bống Kê Cây Xanh         Xanh         Dầu       Image: China Sách Thống Kê         Nam Kỳ Khởi Nghĩa       STT         2008       Image: China Sách Thống Xê         Chẵn       Image: China Sách Thống Xê         Chẵn       Image: China Sách Thống Xê         Xuất Báo Cáo       STT

Hình 3.41: Form Báo cáo – Thống kê cây xanh

Form Báo cáo – Thống kê cây xanh gồm các thành phần sau:

- Các Label: Tên Loại Cây, Tên Đường, Năm Trồng, Lề Đường và các Combobox tương ứng thể hiện các giá trị của từng label.
- jScrollPane thể hiện toàn bộ danh sách thống kê khi ta kích chọn nút "Thống Kê".
- Hai button: "Thống Kê" và "Xuất Báo Cáo".



Hình 3.42: Tạo file ThongKe.java thể hiện form Báo Cáo – Thống Kê Cây Xanh

Nội dụng file ThongKe.java sẽ được trình bày trong phần phụ lục 2.

3.4.3. Tạo cửa sổ About giới thiệu project



Hình 3.43: Cửa sổ About giới thiệu project QLCX

Trong hệ thống gvSIG, cửa sổ About (menu Help/About) giới thiệu về gvSIG. Ngoài ra, gvSIG cho phép các extension gắn phần giới thiệu của mình vào cửa sổ này. Mỗi extension sẽ được giới thiệu trên 1 tabpage. Nội dung trên tabpage được định dạng như trang html.

#### 3.4.3.1. Tạo trang html

Tạo thư mục help trong thư mục extQLCX. Sau đó tạo file about.htm và file hình ảnh đi kèm được tham chiếu trong file about.html



Hình 3.44: Thư mục help chứa file about giới thiệu project

Nội dung file about.htm như sau:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
 <title>QLCX Phường 6,Quân 3,TPHCM</title>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
 <style></style>
</head>
<body style="width: 414px; height: 199px">
 <b><font face="Geneva, Arial,</pre>
Helvetica, san-serif"><h1>QLCX.P6.Q3</h1></font></b><img style="width:
                                                                       150px;
height: 116px; color: Black" alt="" src="about.jpg">
<br><font face="Geneva, Arial, Helvetica, san-serif"> Hoàn Thành bởi Nguyễn Quốc
Tuán</font> <font face="Geneva, Arial, Helvetica, san-serif"><a href="Lớp DH07GI,
Trường Đại Học Nông Lâm TPHCM" </a></font></body>
</html>
```

## 3.4.3.2. Tạo extension thể hiện about

Tạo file AboutExtension.java trong thư mục src.



Hình 3.45: Tạo file AboutExtension.java thể hiện about

Nội dung file AboutExtension.java như sau:

```
package glcx;
import com.iver.andami.PluginServices;
import com.iver.andami.plugins.Extension;
import com.iver.cit.gvsig.About;
import com.iver.cit.gvsig.gui.panels.FPanelAbout;
public class AboutExtension extends Extension {
       public boolean isEnabled() {
              return false;
       }
       public boolean isVisible() {
              return false;
       }
       public void initialize() {
       public void postInitialize(){
              About about = (About) PluginServices.getExtension(About.class);
              FPanelAbout panelAbout = about.getAboutPanel();
              java.net.URL
                                                      aboutURL
AboutExtension.class.getResource("/help/about.htm");
              panelAbout.addAboutUrl("QLCX", aboutURL);
       }
```

=

```
public void execute(String actionCommand) {
    }
public void initialize1() {
    }
}
```

## 3.4.4. Tạo bộ cài đặt bằng IzPack



Hình 3.46: Cửa sổ tiến hành cài đặt project QLCX

IzPack (http://izpack.org/) là công cụ hỗ trợ đóng gói và triển khai phần mềm. Tiến hành cài đặt IzPack.

Tạo thư mục install trong thư mục extQLCX chứa 2 file build.xml và install.xml và thư mục con là resources.



Hình 3.47: Thư mục install trong thư mục extQLCX

Nội dung file build.xml như sau:

```
<project name="instalation-qvSIG" default="distribution" basedir=".">
       <!-- <property name="base_dir" location="./"/>
       <property name="dir_libs" location="/home/volaya/soft/izpack/lib"/>
       <property name="vers" location="0.6"/>
       <property name="izpack_dir" location="/home/volaya/soft/izpack"/> -->
       <property name="base_dir" location="./"/>
       <property name="dir_libs" location="C:\Program Files\IzPack\lib"/>
       <property name="vers" location="0.6"/>
       <property name="izpack_dir" location="C:\Program Files\IzPack"/>
       <taskdef name="izpack" classpath="${dir_libs}/compiler.jar"
classname="com.izforge.izpack.ant.IzPackTask"/>
       <target name="distribution"
             depends="generate">
       </target>
       <target name="generate">
             <izpack input="install.xml"
                           output="QLCX_P6Q3Setup"
                           installerType="standard"
                           basedir="${base dir}"
```

</target> </project>

Nội dung file install.xml như sau:

```
<installation version="1.0">
      <info>
             <appname>QLCX_P6Q3TPHCM</appname>
             <appversion>1.0.0</appversion>
             <authors>
                    <author name="SINH VIÊN: NGUYÊN QUỐC TUẤN" email="" />
             </authors>
             <javaversion>1.5</javaversion>
      </info>
       <guiprefs height="430" resizable="no" width="800">
             <laf name="liquid">
                    <os family="unix" />
             </laf>
             <laf name="liquid">
                    <os family="windows" />
             </laf>
      </quiprefs>
      <locale>
             <!-- <langpack iso3="vne"/> -->
             <langpack iso3="eng"/>
      </locale>
       <resources>
             <res src="resources/gpl.txt" id="LicencePanel.licence"/>
             <res src="resources/readme.htm" id="HTMLInfoPanel.info eng"/>
             <res src="resources/glcx.jpg" id="Installer.image"/>
             <res src="resources/userInputSpec.xml" id="userInputSpec.xml"/>
     <parsable os="linux" targetfile="resources/userInputSpec.xml"/>
    <res src="resources/userInputLang.xml_eng" id="userInputLang.xml_eng" />
      </resources>
      <panels>
             <panel classname="HTMLInfoPanel"/>
             <!-- <panel classname="LicencePanel"/> -->
             <panel classname="UserInputPanel"/>
             <panel classname="InstallPanel"/>
             <!-- <panel classname="SimpleFinishPanel"/> -->
      </panels>
       <packs>
             <pack name="QLCX_P6Q3TPHCM" required="yes">
                    <description>QLCX_P6Q3TPHCM</description>
                    <file targetdir="$INSTALL PATH/bin/gvSIG/extensiones/"
src="../../_fwAndami/gvSIG/extensiones/qlcx" override="true"/>
             </pack>
       </packs>
</installation>
```

#### 3.5. Chức năng của hệ thống

Chức năng của hệ thống gồm: cập nhật cây xanh và xem báo cáo thống kê. Cập nhật thì sử dụng công cụ sẵn có của gvSIG, báo cáo thống kê thì dùng extQLCX.

## 3.5.1. Công cụ cập nhật dữ liệu sử dụng công cụ sẵn có trong gvSIG

### 3.5.1.1. Công cụ thêm điểm

Để thêm điểm, sử dụng công cụ Point hoặc Multipoint



Hình 3.48:Trước khi sử dụng công cụ point để thêm điểm



Hình 3.49: Sau khi sử dụng công cụ point để thêm điểm với các điểm màu đỏ

#### 3.5.1.2. Công cụ chỉnh sửa Copying



Pr 41

Để sao chép một đối tượng trong gvSIG, sử dụng công cụ

Hình 3.50: Đối tượng dạng đường trước khi sử dụng công cụ Copying



Hình 3.51: Sử dụng công cụ Copying sao chép đối tượng dạng đường

## 3.5.1.3. Công cụ thêm đối tượng dạng đường



Để thêm đối tượng dạng đường, sử dụng công cụ Line

Hình 3.52: Trước khi sử dụng công cụ Line để thêm đối tượng dạng đường



Hình 3.53: Sau khi sử dụng công cụ Line thêm đối tượng dạng đường

## 3.5.1.4. Sử dụng bảng thuộc tính chỉnh sửa đối tượng

1/ Thêm một record

Để thêm một record mới, một dòng trắng sẽ xuất hiện trong bảng thuộc tính.

	🚭 Table: A	ttribute tabl	e: VNM		×
	Country	ADMName	ADMUnitID	ADMPar	
	Vietnam	Soc Trang	70410124	70409614	^
	Vietnam	Long An	70410230	70409614	
	Vietnam	Bac Lieu	70410428	70409614	
	Vietnam	Ca Mau	70410485	70409614	:
	Vietnam	Dong Thap	70410569	70409614	:
	Vietnam	An Giang	70410720	70409614	
il 4 🔪 🧹	Vietnam	Tien Giang	70410872	70409614	
	Vietnam	Vinh Long	70411045	70409614	
					~
	<			>	
	1 / 62 Total red	ords selected.			

Hình 3.54: Dòng trắng xuất hiện khi thêm một record mới

#### 2/ Chỉnh sửa record

Để sửa đổi record của một đối tượng, chọn đối tượng và đi đến bảng thuộc tính, khi đó record sẽ sáng lên tương ứng với đối tượng được chọn. Kích chọn cột cần chỉnh sửa trong record.

🎯 Table: /	Attribute tab	le: VNM		
Country	ADMName	ADMUnitID	ADMPar	1 1
Vietnam	Soc Trang	70410124	70409614	
Vietnam	Long An	70410230	70409614	<b></b>
Vietnam	Bac Lieu	70410428	70409614	2
Vietnam	Ca Mau	70410485	70409614	
Vietnam	Dong Thap	70410569	70409614	
Vietnam	An Giang	70410720	70409614	
Vietnam	Tien Giang	70410872	70409614	
Vietnam	Vinh Long	70411045	70409614	
			~	
< -		÷	>	
1 / 62 Total re	ecords selected.			

Hình 3.55: Sửa record trong bảng thuộc tính

#### 3/ Xóa bỏ record

Để xóa bỏ record của một đối tượng, chọn đối tượng và đi đến bảng thuộc tính (hay chọn trực tiếp trong bảng thuộc tính), đến menu Table  $\rightarrow$  Remove row.



Hình 3.56: Xóa bỏ record

3.5.2. Form Báo cáo – Thống kê cây xanh

🎯 Báo Cáo - T	hống Kê Cây Xanh									×
⊂Thống Kê Cây)	Thống Kê Cây Xanh									
Tên Loại Cây	Däu	~	Danh	Sách T	hống Kế	è				
-0 - 0	River IVA IVE 24 RIVE?		STT	Sõ	Тê	Τê	Nă	Lê	Gh	
Tên Đường	Nam Ky Khơi Nghia	<b>*</b>	1	190	Nam	Dầu	2008	Chẵn		<u>~</u>
Năm Trồng	2008	~	2	186	Nam	Dầu	2008	Chẵn		
Nam Hong		·	3	196A	Nam	Dầu	2008	Chẵn		
Lê Đườna	Chẵn	~	4	196A	Nam	Dầu	2008	Chẵn		
			5	208	Nam	Dầu	2008	Chẵn		
		_	6	218	Nam	Dầu	2008	Chẵn		
Thống Kê	Xuất Báo Cáo		7	208	Nam	Dầu	2008	Chẵn		<b>~</b>

Hình 3.57: Form Báo cáo – Thống kê cây xanh

Người dùng có thể chọn các giá trị cần thể hiện như tên loại cây, tên đường, năm trồng, hay lề đường trong "Danh sách thống kê". Sau đó nhấn chọn nút "Thống Kê", một danh sách sẽ hiện ra theo các giá trị người dùng đã chọn. Nếu muốn xuất danh sách trong "Danh sách thống kê" ra bảng báo cáo, người dùng nhấn chọn nút "Xuất Báo Cáo", một biểu mẫu báo cáo tương ứng với danh sách trong "Danh sách thống kê" sẽ được xuất ra. Sử dụng công cụ hỗ trợ JasperReports là thư viện tạo báo cáo, iReport là công cụ hỗ trợ thiết kế báo cáo trên nền JasperReports một cách trực quan.

# Chương 4

# KẾT LUẬN VÀ KIẾN NGHỊ

## 4.1. Kết luận

Đề tài đã đạt được các kết quả sau:

- Nghiên cứu công cụ mã nguồn mở gvSIG trong việc phục vụ xây dựng ứng dụng GIS cụ thể là trong quản lý cây xanh.
- Tìm hiểu và ứng dụng các công cụ sẵn có trong phân tích không gian của phần mềm gvSIG.
- Xây dựng một plugin QLCX phục vụ cho việc báo cáo và thống kê cây xanh.

Đề tài sử dụng phần mềm GIS mã nguồn mở gvSIG và các phần mềm mã nguồn mở khác như hệ quản trị CSDL PostgreSQL, môi trường lập trình eclipse trong Java xây dựng ứng dụng có ý nghĩa thiết thực và hỗ trợ tốt cho người dùng.

Sử dụng các phần mềm mã nguồn mở giúp tiết kiệm chi phí bản quyền so với các phần mềm thương mại cũng như có thể truy cập vào mã nguồn của chương trình.

## 4.2. Kiến nghị

Tiếp tục nghiên cứu hoàn thiện hệ thống quản lý cây xanh, xây dựng chức năng cập nhật dữ liệu không gian, chức năng tìm kiếm, chức năng bảo dưỡng cây xanh.

Tiếp tục nghiên cứu các dự án có sử dụng phần mềm gvSIG để học hỏi và phát triển gvSIG vào những mục đích cụ thể khác.

Tiếp tục nghiên cứu các phần mềm mã nguồn mở khác.

# TÀI LIỆU THAM KHẢO

- Quách Đồng Thắng, 2008. Nghiên cứu xây dựng ứng dụng GIS bằng phần mềm mã nguồn mở. Áp dụng xây dựng ứng dụng quản lý vi phạm trật tự xây dựng tại P.Hiệp Bình Phước – Q.Thủ Đức – TP.HCM. Luận văn Thạc sĩ chuyên ngành Bản đồ, viễn thám và hệ thống thông tin địa lý, Đại Học Quốc Gia TP.HCM, Trường ĐH Bách Khoa.
- Phòng Kỹ Thuật, 2010. *Tài liệu hướng dẫn lập trình gvSIG*. Trung tâm ứng dụng Hệ thống thông tin địa lý, Sở Khoa Học – Công Nghệ TP.HCM.
- 3. IVER Tecnologías de la información (oficinas centrales en Valencia), 2007. gvSIG
   User guide. Generalitat Valenciana.
- <u>http://www.hochiminhcity.gov.vn/default.aspx</u>. Truy cập ngày 15 tháng 6 năm 2011.
- 5. <u>http://www.gvsig.gva.es/</u>. Truy cập ngày 25 tháng 4 năm 2011.
- 6. <u>http://postgis.refractions.net/</u>. Truy cập ngày 25 tháng 4 năm 2011.

## PHŲ LŲC

## Phụ lục 1. Hướng dẫn cài đặt công cụ QLCX

Công cụ QLCX là phần mở rộng được viết trên nền tảng gvSIG 1.9.

#### 1. Cài đặt gvSIG 1.9

Cài đặt theo trình tự:

- JRE 1.6.x (tåi tại trang web: http://www.oracle.com/technetwork/java/javase/downloads/index.html).
- JAI 1.1.x for JRE (tåi tại trang web: <u>http://jai.dev.java.net/binany-builds.html</u>).
- JAI Image I/O 1.x for JRE (tåi tai trang web: <u>http://jai-imageio.dev.java.net/binany-builds.html</u>).
- gvSIG 1.9 (tåi tại trang web: <u>http://www.gvsig.org/web/</u>).

#### 2. Cài đặt công cụ QLCX

Chạy file QLCX\_P6Q3Setup.jar, chọn kext để tiếp tục cài đặt





Chọn đường dẫn tới thư mục cài đặt gvSIG 1.9 (mặc định là C:\Program Files\gvSIG 1.9).



Chọn đường dẫn tới folder cài đặt gvSIG

Chọn đường dẫn tới thư mục cài đặt gvSIG

Chọn 🕪 Next để tiếp tục.

Cài đặt xong thì bấm nút Quit

Khởi động gvSIG, người dùng sẽ thấy menu QLCX



Phần mở rộng QLCX trong gvSIG

## 3. Cài đặt hệ quản trị cơ sở dữ liệu PostgreSQL

Người dùng chạy file "postgresql-8.4.6-1-windows.exe" (được tải tại trang web: <u>http://www.postgresql.org/</u>) để tiến hành cài đặt.

Tiếp theo người dùng chạy file "postgis\_1\_5\_pg84.exe" để cài đặt extension PostGIS (tải tại trang web: <u>http://postgis.refractions.net/</u>).

Sau khi cài đặt PostgreSQL 8.4 và phần mở rộng PostGIS (hỗ trợ lưu dữ liệu không gian), extension PostGIS sẽ tạo ra một PostGIS database template với tên là template\_postgis, với các table, function... sẵn sàng hỗ trợ quản lý dữ liệu không gian.



Cấu trúc template\_postgis

Sau đó chọn biểu tượng pgAdmin III cơ sở dữ liệu cho ứng dụng.



để khởi động PostgreSQL và tạo



Giao diện pgAdmin III của PostgreSQL

Nhấn phải chuột tại "Database", chọn New Databases để tạo một cơ sở dữ liệu hoàn toàn mới hỗ trợ lưu cơ sở dữ liệu không gian.



Giao diện tạo mới Database

Đặt tên cơ sở dữ liệu là QLCX, mục Owner chọn mặc định là Postgres, mục Template chọn template\_postgis, các phần còn lại để mặc định. Nhấn OK để tạo cơ sở dữ liệu.

🔋 New Database	🔀
Properties Variabl	es Privileges SQL
Name	QLCX
OID	
Owner	postgres 💌
Encoding	UTF8
Template	template_postgis
Tablespace	pg_default 💌
Schema restriction	
Collation	<b>~</b>
Character type	<b>~</b>
Connection Limit	-1
Comment	
Help	OK Cancel

Cấu trúc tạo Database

Sau khi tạo cơ sở dữ liệu QLCX, nhấn phải chuột trên Database và chọn Restore...



Restore Database vào CSDL

Chọn file qlcx.backup đã tạo sẵn. Nhấn OK.
Sau khi restore thành công, các bảng được tạo trong mục Tables



Cấu trúc bảng trong CSDL

## Phụ lục 2. Nội dụng file ThongKe.java

package qlcx.gui;

import java.awt.GridBagLayout; import javax.swing.JPanel; import javax.swing.BorderFactory; import javax.swing.border.TitledBorder; import javax.swing.table.DefaultTableCellRenderer; import javax.swing.table.DefaultTableModel;

import java.awt.Cursor; import java.awt.Font; import java.awt.Color; import java.awt.Dimension; import java.awt.Dimension; import java.awt.Rectangle; import java.awt.Rectangle; import javax.swing.JComboBox; import javax.swing.JButton; import javax.swing.JDptionPane; import javax.swing.JScrollPane; import javax.swing.JTable; import javax.swing.ListSelectionModel;

import com.iver.andami.ui.mdiManager.IWindow; import com.iver.andami.ui.mdiManager.IWindowListener; import com.iver.andami.ui.mdiManager.WindowInfo; import com.iver.cit.gvsig.fmap.drivers.ConnectionFactory; import com.iver.cit.gvsig.fmap.drivers.ConnectionJDBC; import com.iver.cit.gvsig.fmap.drivers.DBException; import com.iver.cit.gvsig.fmap.drivers.IConnection;

import java.awt.Point; import java.sql.ResultSet; import java.sql.Statement; import java.util.HashMap; import java.util.Vector;

import qlcx.ConnecttoPostgis; import qlcx.MyReportViewer; public class ThongKe extends JPanel implements IWindow, IWindowListener {

```
private static final long serialVersionUID = 1L;
private JLabel jLabel = null;
private JLabel jLabel1 = null;
private JLabel jLabel2 = null;
private JLabel jLabel3 = null;
private JComboBox cboLC = null;
private JComboBox cboTD = null;
private JComboBox cboNT = null;
private JComboBox cboLD = null;
private JButton Btxbc = null;
private JButton ThongKe = null;
private JScrollPane jScrollPane = null;
private JTable tblTK = null;
private IConnection conn = null;
private ConnecttoPostgis Connectpostgis = null; // @jve:decl-index=0:
private String dburl = ""
private String dbuser = ""; // @jve:decl-index=0:
private String dbpass = "";
private int count;
private int rowupdated = 0;
private int countlc;
private String [][] loaicay;
private int counttd;
```

```
private String [][] tuyenduong;
         private int countld;
        private String [][] leduong;
        private int countnt;
        private String [][] namtrong;
        public ThongKe() {
                 super();
                 initialize();
                 ConnectDB();
                 getlc();
                 gettd();
                 getnt();
                 getld();
        private void initialize() {
                 jLabel3 = new JLabel();
                 jLabel3.setText("Le Duorg");
                 jLabel3.setSize(new Dimension(80, 25));
                 jLabel3.setLocation(new Point(16, 119));
                 jLabel2 = new JLabel();
                 jLabel2.setText("Năm Trồng");
                 jLabel2.setSize(new Dimension(80, 25));
                 jLabel2.setLocation(new Point(18, 89));
                 jLabel1 = new JLabel();
                 jLabel1.setText("Tên Đường");
                 jLabel1.setSize(new Dimension(80, 25));
                 jLabel1.setLocation(new Point(17, 60));
                 jLabel = new JLabel();
                 jLabel.setText("Tên Loai Cây");
                 jLabel.setSize(new Dimension(80, 25));
                 jLabel.setLocation(new Point(17, 31));
                 this.setSize(567, 243);
                 this.setLayout(null);
                                                                             "Thống
                                                                                                   Cây
                 this.setBorder(BorderFactory.createTitledBorder(null,
                                                                                          Kê
                                                                                                             Xanh".
TitledBorder.DEFAULT JUSTIFICATION, TitledBorder.DEFAULT POSITION, new Font("Dialog", Font.BOLD, 12), new
Color(51, 51, 51)));
                 this.add(jLabel, null);
                 this.add(jLabel1, null);
                 this.add(jLabel2, null);
                 this.add(jLabel3, null);
                 this.add(cboLC(), null);
                 this.add(cboTD(), null);
                 this.add(cboNT(), null);
                 this.add(cboLD(), null);
                 this.add(Btxbc(), null);
                 this.add(ThongKe(), null);
                 this.add(getJScrollPane(), null);
        }
        public void ConnectDB() {
                 Connectpostgis = new ConnecttoPostgis();
                 dburl = Connectpostgis.Connectqlcx()[0];// "jdbc:postgresql://192.168.8.107/PCCC";
                 dbuser = Connectpostgis.Connectqlcx()[1];
                 dbpass = Connectpostgis.Connectqlcx()[2];
                 try {
                          Class.forName("org.postgresql.Driver");
                          conn = ConnectionFactory.createConnection(dburl, dbuser, dbpass);
                 } catch (Exception e) {
                          e.printStackTrace();
                          JOptionPane.showMessageDialog(null, e.toString());
                 }
        }
        public void getlc()
         {
                 cboLC.removeAllItems();
                          try{
```

```
String SQLString;
                                    Statement st = ((ConnectionJDBC) conn).getConnection()
                                    .createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                                      ResultSet.CONCUR_READ_ONLY); //
                                    SQLString = "select distinct loai cay from cayxanh shp order by loai cay ";
                                    ResultSet rs = st.executeQuery(SQLString);
                                    rs.last();
                           countlc=rs.getRow();
                           loaicay = new String [2][countlc];
                                    rs.beforeFirst();
                                    int id=0;
                                    while( rs.next() )
                                     {
                                             cboLC.addItem(rs.getString("loai_cay").toString());
loaicay[0][id]= rs.getString("loai_cay");
loaicay[1][id]= rs.getString("loai_cay");
                                             st.close();
                                     }
                           catch (Exception e){
                              e.printStackTrace();
                              JOptionPane.showMessageDialog(null,e.toString());
                           } }
        public String getlc1()
         {
                  String malc = null;
                             for (int k = 0; k < \text{countle}; k++)
                             {
                                      if (cboLC.getSelectedItem().toString().equals(loaicay[1][k].toString()))
                                      {
                                               malc = loaicay[0][k];
                                      }
                                                         }
                           return malc;
        }
        public void gettd()
                  cboTD.removeAllItems();
                           try{
                                    String SQLString;
                                    Statement st = ((ConnectionJDBC) conn).getConnection()
                                    .createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                                      ResultSet.CONCUR_READ_ONLY); //
                                    SQLString = "select distinct ten_duong from cayxanh_shp order by ten_duong
";
                                    ResultSet rs = st.executeQuery(SQLString);
                                    rs.last();
                           counttd=rs.getRow();
                                    tuyenduong = new String [2][counttd];
                                    rs.beforeFirst();
                                    int id=0;
                                    while( rs.next() )
                                     {
                                             cboTD.addItem(rs.getString("ten_duong").toString());
                                             tuyenduong[0][id]= rs.getString("ten_duong");
                                             tuyenduong[1][id] = rs.getString("ten_duong");
                                             st.close();
                                     }
                           catch (Exception e){
                              e.printStackTrace();
                              JOptionPane.showMessageDialog(null,e.toString());
                                                          65
```

```
} }
public String gettd1()
        String matd = null;
                   for (int k = 0; k < \text{counttd}; k++)
                   {
                            if (cboTD.getSelectedItem().toString().equals(tuyenduong[1][k].toString()))
                            {
                                    matd = tuyenduong[0][k];
                            }
                                             }
                 return matd;
public void getId()
        cboLD.removeAllItems();
                 try{
                          String SQLString;
                          Statement st = ((ConnectionJDBC) conn).getConnection()
                          .createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                           ResultSet.CONCUR_READ_ONLY); //
                          SQLString = "select distinct le from cayxanh_shp order by le ";
                          ResultSet rs = st.executeQuery(SQLString);
                          rs.last();
                 countId=rs.getRow();
                          leduong = new String [2][countId];
                          rs.beforeFirst();
                          int id=0;
                          while( rs.next() )
                          {
                                  cboLD.addItem(rs.getString("le").toString());
                                  leduong[0][id]= rs.getString("le");
                                  leduong[1][id]= rs.getString("le");
                                  st.close();
                          }
                  catch (Exception e){
                    e.printStackTrace();
                    JOptionPane.showMessageDialog(null,e.toString());
                 } }
public String getId1()
{
        String mald = null ;
                   for (int k = 0; k < countld; k++)
                   {
                            if (cboLD.getSelectedItem().toString().equals(leduong[1][k].toString()))
                            {
                                    mald = leduong[0][k];
                            }
                                             }
                 return mald;
public void getnt()
ł
        cboNT.removeAllItems();
                 try{
                          String SQLString;
                          Statement st = ((ConnectionJDBC) conn).getConnection()
                          .createStatement(ResultSet.TYPE SCROLL INSENSITIVE,
                                           ResultSet.CONCUR READ ONLY); //
                          SQLString = "select distinct nam_trong from cayxanh_shp order by nam_trong
                          ResultSet rs = st.executeQuery(SQLString);
```

rs.last();

";

```
66
```

```
countnt=rs.getRow();
                                      namtrong = new String [2][countnt];
                                      rs.beforeFirst();
                                      int id=0:
                                      while( rs.next() )
                                       {
                                                cboNT.addItem(rs.getString("nam_trong").toString());
                                                namtrong[0][id] = rs.getString("nam_trong");
                                                namtrong[1][id] = rs.getString("nam_trong");
                                               id++; }
                                       st.close();
                                       }
                             catch (Exception e){
                                e.printStackTrace();
                                JOptionPane.showMessageDialog(null,e.toString());
                            } }
         public String getnt1()
         {
                   String mant = null ;
                               for (int k = 0; k < \text{countnt}; k++)
                               {
                                        if (cboNT.getSelectedItem().toString().equals(namtrong[1][k].toString()))
                                        {
                                                  mant = namtrong[0][k];
                                        }
                                                           }
                            return mant;
         }
         public void TK() {
         String lc = cboLC.getSelectedItem().toString();
          String td = cboTD.getSelectedItem().toString();
          String nt = cboNT.getSelectedItem().toString();
          String Id = cboLD.getSelectedItem().toString();
                   DefaultTableModel dm = (DefaultTableModel) tblTK.getModel();
                   dm.getDataVector().removeAllElements();
                   Vector rowHeader = new Vector();
                   rowHeader.add("STT");
                   rowHeader.add("Số Nhà");
                   rowHeader.add("Tên Đường");
rowHeader.add("Tên LC");
rowHeader.add("Năm Trồng");
                   rowHeader.add("Le");
                   rowHeader.add("Ghi Chú");
                   DefaultTableModel model = new DefaultTableModel(rowHeader, 0);
            tblTK.setModel(model);
         try {
                    String SQLString;
                   Statement st = ((ConnectionJDBC) conn).getConnection()
                                      .createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                                         ResultSet.CONCUR_READ_ONLY); //
         SQLString = "select so_nha, ten_duong, loai_cay, nam_trong, le , ghi_chu from cayxanh_shp where ="" + lc + "'and ten_duong ='" + td + "' and nam_trong ='" + nt + "' and le ='" + ld + "'order by
loai_cay ='"
loai_cay";
         ResultSet rs = st.executeQuery(SQLString);
         rs.last();
         count = rs.getRow();
         rs.beforeFirst();
                   int id=0;
                   while (rs.next()) {
                            Vector rowData:
                   rowData = new Vector();
                            rowData.add(id + 1);
                            rowData.add(rs.getString("so_nha"));
                            rowData.add(rs.getString("ten_duong"));
rowData.add(rs.getString("loai_cay"));
```

```
rowData.add(rs.getString("nam_trong"));
                rowData.add(rs.getString("le"));
                rowData.add(rs.getString("ghi_chu"));
                model.addRow(rowData);
                id++;
        }
        st.close();
        RightTableCellRenderer
                                  aRightTableCellRenderer
                                                                             RightTableCellRenderer();
                                                                    new
                                                               =
tblTK.getColumnModel().getColumn(0).setCellRenderer(aRightTableCellRenderer);
        tblTK.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
        tblTK.setSelectionMode(ListSelectionModel.SINGLE SELECTION );
        //tblTK.getColumnModel().getColumn(0).setCellRenderer(new CellRenderer());
        if (tblTK.getRowCount()>0)
        {
                 tblTK.requestFocus();
                tblTK.setRowSelectionInterval(rowupdated,rowupdated);
        } catch (Exception e1) {
        e1.printStackTrace();
        JOptionPane.showMessageDialog(null,e1);
                }
class RightTableCellRenderer extends DefaultTableCellRenderer {
          protected RightTableCellRenderer() {
           setHorizontalAlignment(JLabel.CENTER); }
}
private JComboBox cboLC() {
        if (cboLC == null) {
                cboLC = new JComboBox();
                cboLC.setLocation(new Point(93, 32));
                cboLC.setSize(new Dimension(140, 25));
        }
        return cboLC;
}
private JComboBox cboTD() {
        if (cboTD == null) {
                cboTD = new JComboBox();
                cboTD.setLocation(new Point(95, 59));
                cboTD.setSize(new Dimension(140, 25));
        return cboTD;
private JComboBox cboNT() {
        if (cboNT == null) {
                cboNT = new JComboBox();
                 cboNT.setLocation(new Point(95, 88));
                cboNT.setSize(new Dimension(140, 25));
        }
        return cboNT;
private JComboBox cboLD() {
        if (cboLD == null) {
                cboLD = new JComboBox();
                cboLD.setLocation(new Point(96, 118));
                cboLD.setSize(new Dimension(140, 25));
        }
        return cboLD;
}
private JButton Btxbc() {
        if (Btxbc == null) {
                 Btxbc = new JButton();
                 Btxbc.setPreferredSize(new Dimension(100, 25));
                 Btxbc.setLocation(new Point(121, 164));
                 Btxbc.setSize(new Dimension(110, 25));
                 Btxbc.setText("Xuất Báo Cáo");
                Btxbc.addActionListener (new java.awt.event.ActionListener() {
```

}

public void actionPerformed(iava.awt.event.ActionEvent e) { setCursor(Cursor.getPredefinedCursor(Cursor.WAIT\_CURSOR)); String baocao=""; HashMap parameter=new HashMap(); Đâu") if cboLC.getSelectedItem().toString().equals("Bã 88 cboTD.getSelectedItem().toString().equals("Hồ Xuân Hương")&& cboNT.getSelectedItem().toString().equals("2006") && cboLD.getSelectedItem().toString().equals("Lé")) { JasperFile = System.getProperty("user.dir") String + "\\qvSIG\\extensiones\\qlcx\\templates\\report1.jasper"; MyReportViewer viewer=new MyReportViewer("Thống kê cây xanh","jdbc:postgresql://localhost/qlcx", "postgres", "postgres", JasperFile, parameter); viewer.setVisible(true); if (cboLC.getSelectedItem().toString().equals("Bång lăng") 88 else cboTD.getSelectedItem().toString().equals("Cách Mang Tháng Tám")&& cboNT.getSelectedItem().toString().equals("2002") && cboLD.getSelectedItem().toString().equals("Chan")) String System.getProperty("user.dir") JasperFile + = "\\gvSIG\\extensiones\\qlcx\\templates\\report2.jasper"; MyReportViewer viewer=new MyReportViewer("Thống kê cây xanh","jdbc:postgresql://localhost/qlcx", "postgres", "postgres", JasperFile, parameter); viewer.setVisible(true); setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT\_CURSOR)); }}); return Btxbc; } private JButton ThongKe() { if (ThongKe == null) { ThongKe = **new** JButton(); ThongKe.setText("Thống Kê"); ThongKe.setSize(new Dimension(90, 25)); ThongKe.setLocation(new Point(16, 166)); ThongKe.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent e) { try { //ClearSelection("TK"); } catch (Exception e1) { // TODO Auto-generated catch block e1.printStackTrace(); } setCursor(Cursor.getPredefinedCursor(Cursor.WAIT\_CURSOR)); if (cboLC.getSelectedItem().toString() != null && cboTD.getSelectedItem() !=null && cboNT.getSelectedItem() !=null && cboLD.getSelectedItem() !=null) TK(); { setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT\_CURSOR)); } }); } return ThongKe; private JScrollPane getJScrollPane() { if (jScrollPane == null) { jScrollPane = **new** JScrollPane(); jScrollPane.setBounds(new Rectangle(249, 26, 305, 162)); jScrollPane.setBorder(BorderFactory.createTitledBorder(null, "Danh Sách Thống Kê", TitledBorder.DEFAULT JUSTIFICATION, TitledBorder.DEFAULT POSITION, new Font("Dialog", Font.BOLD, 12), new Color(51, 51, 51))); jScrollPane.setViewportView(tblTK()); return jScrollPane;

```
}
private JTable tblTK() {
    if (tblTK == null) {
        tblTK = new JTable();
    }
    return tblTK;
}
@Override
public void windowActivated() {
    @Override
public void windowClosed() {
    @Override
public WindowInfo getWindowInfo() {
        // TODO Auto-generated method stub
        WindowInfo vi = new WindowInfo(WindowInfo.ICONIFIABLE);
        vi.setTitle("Báo Cáo - Thống Kê Cây Xanh");
        return vi;
}
@Override
public Object getWindowProfile() {
        // TODO Auto-generated method stub
        return null;
}
```