Object Oriented Programming Report

Historical information application

Team Members:

Nguyen Thanh Dat - 20215194

Tran Trong Ky - 20194784

Nguyen Nhu Dai - 20215191

Ngo Vinh Khanh - 20215211

Tran Tung Duong - 20215190

Nguyen Dinh Hong Phong - 20215233

Date: July 11, 2023

1 Group introduction

This is the report for the midterm Object Oriented Programming project by group 4 from class Object Oriented Programming - 139411

The project of our group is a desktop application to display information about Viet Nam history. Specifically, the app provides features to dicover historical figures, events, relics, and festivals with ease using the powerful and user-friendly user interface. Futher information about the product will be elaborated below

2 Workload contribution

During collaboration, we believe that workload is distributed in a fair and balanced parts among all members. Therefore each member's contribution should be evenly $\frac{1}{6}$ of the work, or approximately **17%** each member

Here is how we divide the project into small tasks:

- Nguyen Thanh Dat: Develop the frontend and backend of the application, create an user-friendly graphic user interface, develop the functions of the application and create the structure of the project
- Tran Trong Ky: Scrape data about "events" with related parts and drawing the class diagram for the project
- Nguyen Nhu Dai: Scrape data about "figures" with related information (such as "dynasty"), handle encode, decode functions while working with JSON files and write this report
- Ngo Vinh Khanh: Scrape data about "relics", assign them with related "figures" and create package dependency diagram of the project
- Tran Tung Duong: Find other sources to scrape more data about "figures" and "dynasty" and write the report
- Nguyen Dinh Hong Phong: Scrape data about "festival", assign them to related "figure" or "event", . . . and create the package dependency diagram

Please note that all member did contributed in the idea of the project, how it operates and technologies to be used. In the scraping part, mostly the data are scraped from multiple sources and will be handled the duplication afterward

3 Dataset statistic and overview

In general, the data scraped from websites are divided into the following object: Figure, King, Poinsettia, Event, Relic and Festival, with King and Poinsettia are inherited from Figure. The specific number of each data is shown as:

- Figure: a class represents a historical figures, can be a person, a king or a poinsettia
  - The class includes 9 attributes: id, name, otherName, bornYear, deathYear, parents, dynasties, home, desc (for description)
  - Approximately 3000 figures, containing both Kings and Poinsettias
  - Related data: Figure class contains a attribute of the **Dynasty** that figure belongs to and another attibute store their parents (which are string refer to id of other figures, if found)
- Kings & Poinsettias: Extended classes from Figure

  - King class extends from Figures so it inherit 9 attributes above plus 4 more: regnalYear, posthumousName, eraName, heirApparent
  - Poinsettia also inherit from Figure, inherits 9 attributes and add 2 more: king, graduatedYear
  - Approximately 160 kings and 40 poinsettias
  - King and Poinsettia both inherits from Figure so they also have the same related data

- Dynasty: A class represent a dynasty in history
  - Dynasty class includes 6 attributes: id, name, startYear, endYear, desc (for description)
  - Roughly 29 dynasties throughout the history of Viet Nam crawled
  - In the Dynasty class, each dynasty contain a list of **Figures** that belongs to that dynasty
- Festival: A class represent a festival for celebrating a historical event or to memorize figures or heroes
  - Festival class includes 7 attributes: id, name, place, firstTimes, startTime, relatedFigure, decription
  - About 52 common festivals usually known to many people scraped
  - Festival class contains a attribute of a list of related **figures**
- Historical event: A record of an event that happens throughout Vietnam history
  - Event class includes 9 attributes: id, name, time, location, relatedFigure, dynasty, description, relatedFigsDetails, dynastyDetails
  - About 156 widely known events scraped
  - Event class contains 2 attributes for related data. The first one is related **Figures** that took part in or related to the event, the second one the the **Dynasty** when the event happened in
- Relic: Places that has the remaining or relics of a historical event, such as castles, shrines, . . .
  - Relic class contains 8 attributes: id, name, time, location, Figure, description, province, type
  - About 1070 relics scraped to dataset
  - Similar to almost other class, Relic class also contains a attribute of a list of related figures that used to

4 Data sources

In this project, our group scrape data from multiple websites from the internet, most data are from Wikipedia, Nguoi Ke Su and Van Su

- From Wikipedia, we were able to crawl all of the data above, including Figure, Dynasty, Relic, . . .
- Furthermore, we also get Figures, Dynasty and Event from Nguoikesu website
- And finally, Vansu website is also used for scraping more about Figures and Dynasty

From all of the websites above, after dataset is scraped we will handle the duplication removing part so that we can obtained high-quality and precise dataset

5 UML

Below are our project's class diagram and package diagrams

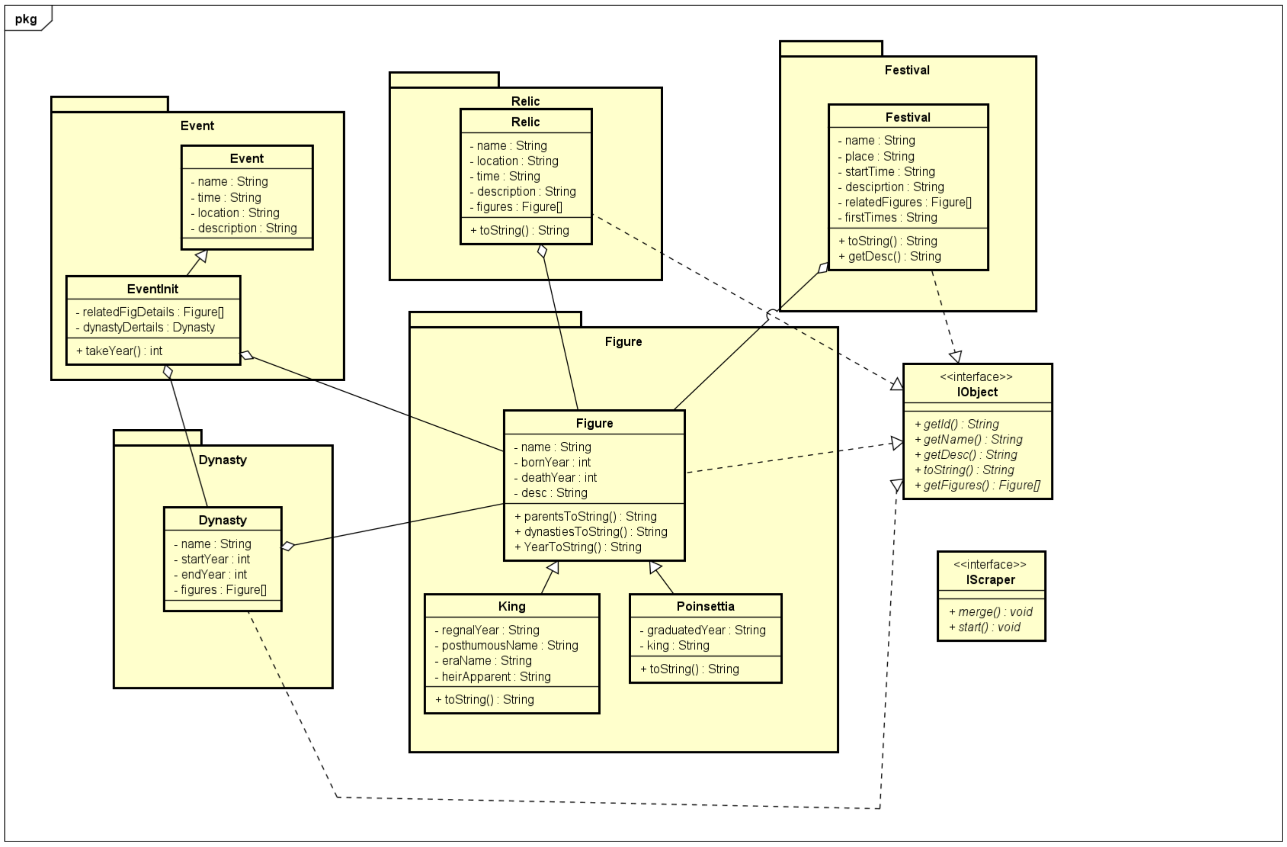


Figure 1: Model's class diagram

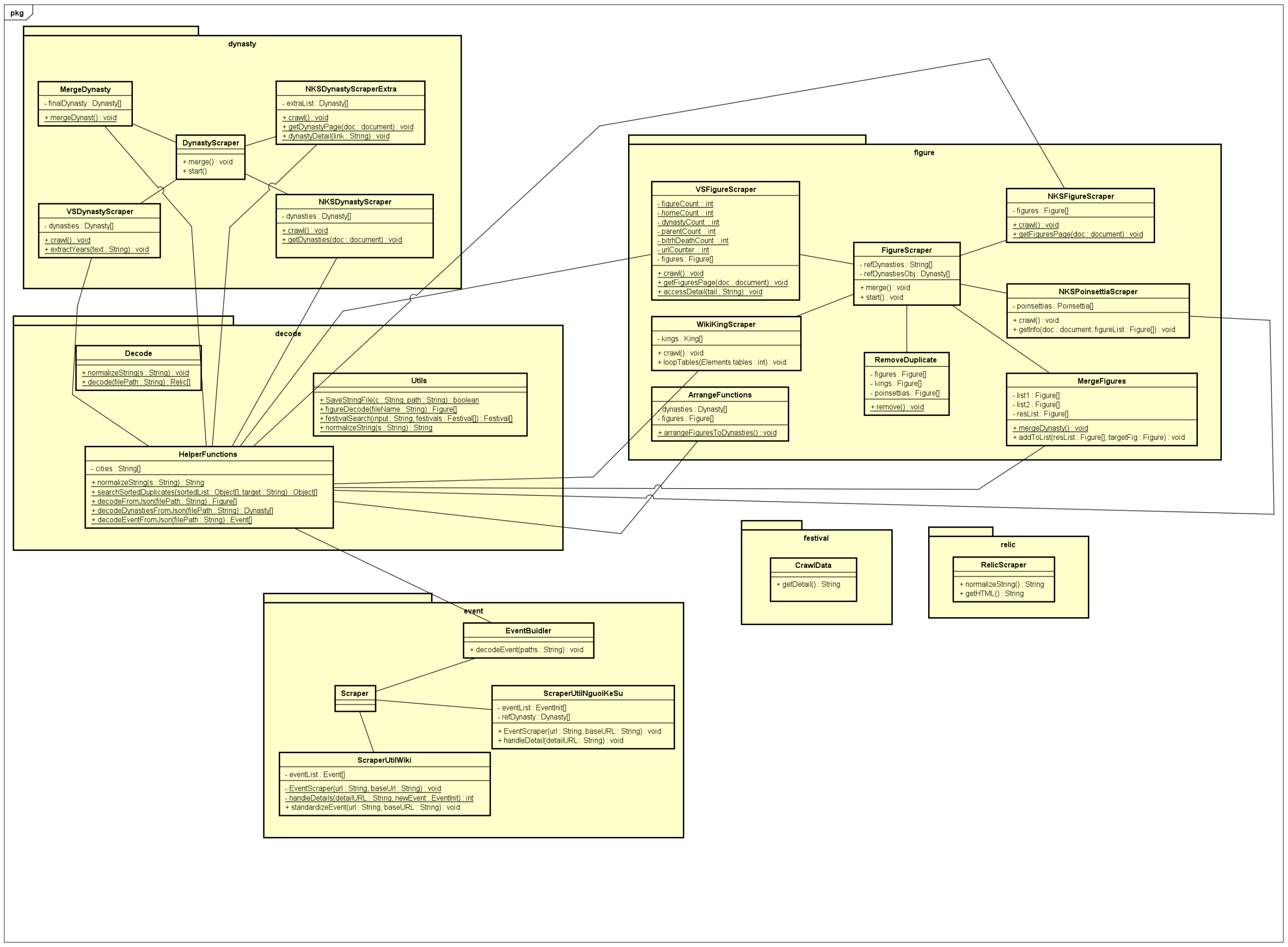

pkg
dynasty
MergeDynasty
- finalDynasty : Dynasty[]
+ mergeDynast() : void
NKSDynastyScraperExtra
- extraList : Dynasty[]
+ crawl() : void
+ getDynastyPage(doc : document) : void
+ dynastyDetail(link : String) : void
DynastyScraper
+ merge() : void
+ start()
VSDynastyScraper
- dynasties : Dynasty[]
+ crawl() : void
+ extractYears(text : String) : void
NKSDynastyScraper
- dynasties : Dynasty[]
+ crawl() : void
+ getDynasties(doc : document) : void
figure
VSFigureScraper
- figureCount : int
- homeCount : int
- dynastyCount : int
- parentCount : int
- bitrhDeathCount : int
- urlCounter : int
- figures : Figure[]
+ crawl() : void
+ getFiguresPage(doc : document) : void
+ accessDetail(tail : String) : void
FigureScraper
- refDynasties : String[]
- refDynastiesObj : Dynasty[]
+ merge() : void
+ start() : void
NKSFigureScraper
- figures : Figure[]
+ crawl() : void
+ getFiguresPage(doc : document) : void
NKSPoinsettiaScraper
- poinsettias : Poinsettia[]
+ crawl() : void
+ getInfo(doc : document, figureList : Figure[]) : void
WikiKingScraper
- kings : King[]
+ crawl() : void
+ loopTables(Elements tables : int) : void
ArrangeFunctions
- dynasties : Dynasty[]
- figures : Figure[]
+ arrangeFiguresToDynasties() : void
RemoveDuplicate
- figures : Figure[]
- kings : Figure[]
- poinsettias : Figure[]
+ remove() : void
MergeFigures
- list1 : Figure[]
- list2 : Figure[]
- resList : Figure[]
+ mergeDynasty() : void
+ addToList(resList : Figure[], targetFig : Figure) : void
decode
Decode
+ normalizeString(s : String) : void
+ decode(filePath : String) : Relic[]
Utils
+ SaveStringFile(c : String, path : String) : boolean
+ figureDecode(fileName : String) : Figure[]
+ festivalSearch(input : String, festivals : Festival[]) : Festival[]
+ normalizeString(s : String) : String
HelperFunctions
- cities : String[]
+ normalizeString(s : String) : String
+ searchSortedDuplicates(sortedList : Object[], target : String) : Object[]
+ decodeFromJson(filePath : String) : Figure[]
+ decodeDynastiesFromJson(filePath : String) : Dynasty[]
+ decodeEventFromJson(filePath : String) : Event[]
festival
CrawlData
+ getDetail() : String
relic
RelicScraper
+ normalizeString() : String
+ getHTML() : String
event
EventBuidler
+ decodeEvent(paths : String) : void
Scraper
ScraperUtilNguoiKeSu
- eventList : EventInit[]
- refDynasty : Dynasty[]
+ EventScraper(url : String, baseURL : String) : void
+ handleDetail(detailURL : String) : void
ScraperUtilWiki
- eventList : Event[]
- EventScraper(url : String, baseUrl : String) : void
- handleDetails(detailURL : String, newEvent : EventInit) : int
+ standardizeEvent(url : String, baseURL : String) : void

Figure 2: Service's class diagram

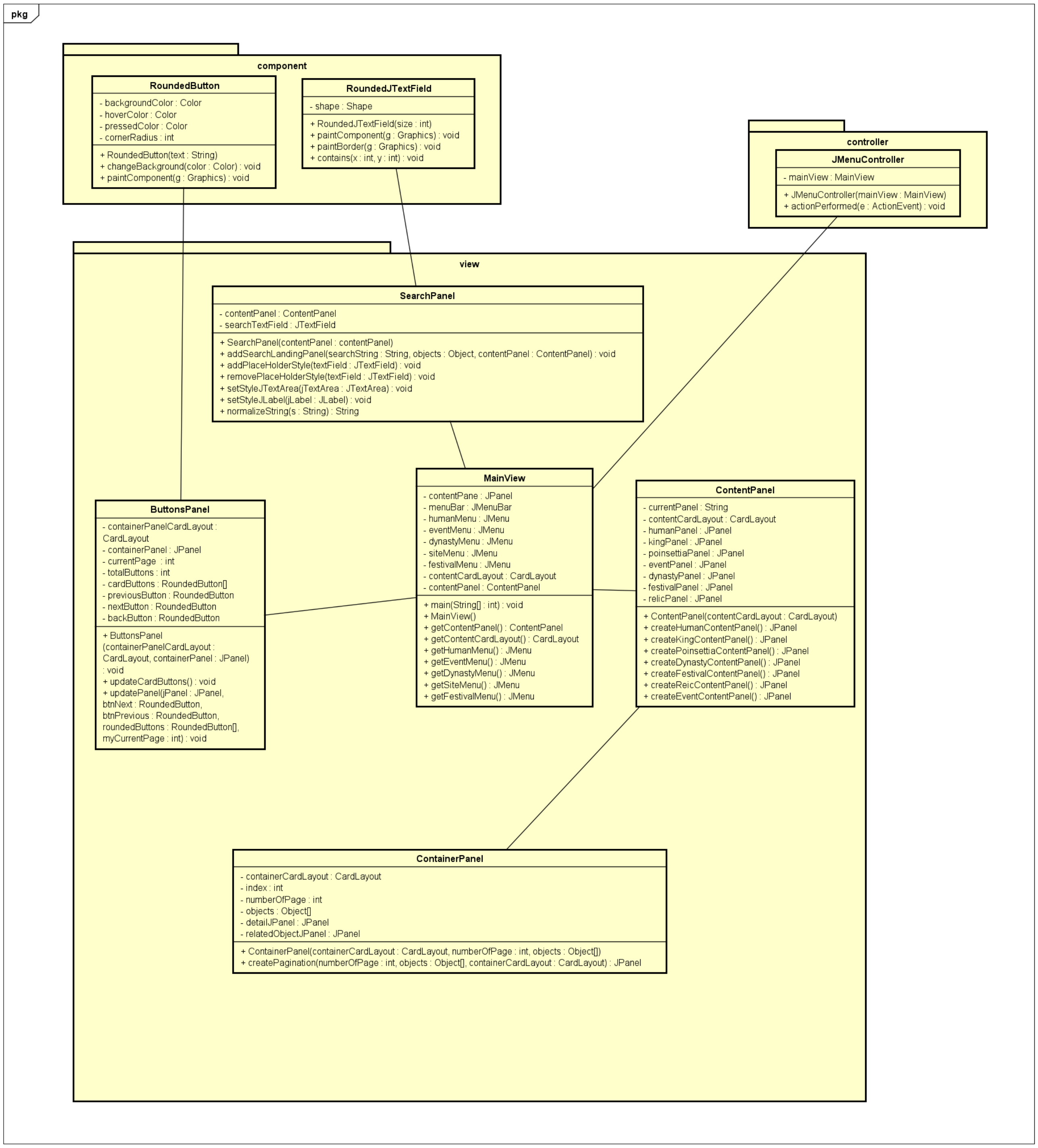

pkg
component
RoundedButton
- backgroundColor : Color
- hoverColor : Color
- pressedColor : Color
- cornerRadius : int
+ RoundedButton(text : String)
+ changeBackground(color : Color) : void
+ paintComponent(g : Graphics) : void
RoundedJTextField
- shape : Shape
+ RoundedJTextField(size : int)
+ paintComponent(g : Graphics) : void
+ paintBorder(g : Graphics) : void
+ contains(x : int, y : int) : void
controller
JMenuController
- mainView : MainView
+ JMenuController(mainView : MainView)
+ actionPerformed(e : ActionEvent) : void
view
SearchPanel
- contentPanel : ContentPanel
- searchTextField : JTextField
+ SearchPanel(contentPanel : contentPanel)
+ addSearchLandingPanel(searchString : String, objects : Object, contentPanel : ContentPanel) : void
+ addPlaceHolderStyle(textField : JTextField) : void
+ removePlaceHolderStyle(textField : JTextField) : void
+ setStyleJTextArea(jTextArea : JTextArea) : void
+ setStyleJLabel(jLabel : JLabel) : void
+ normalizeString(s : String) : String
MainView
- contentPane : JPanel
- menuBar : JMenuBar
- humanMenu : JMenu
- eventMenu : JMenu
- dynastyMenu : JMenu
- siteMenu : JMenu
- festivalMenu : JMenu
- contentCardLayout : CardLayout
- contentPanel : ContentPanel
+ main(String[] : int) : void
+ MainView()
+ getContentPanel() : ContentPanel
+ getContentCardLayout() : CardLayout
+ getHumanMenu() : JMenu
+ getEventMenu() : JMenu
+ getDynastyMenu() : JMenu
+ getSiteMenu() : JMenu
+ getFestivalMenu() : JMenu
ContentPanel
- currentPanel : String
- contentCardLayout : CardLayout
- humanPanel : JPanel
- kingPanel : JPanel
- poinsettiaPanel : JPanel
- eventPanel : JPanel
- dynastyPanel : JPanel
- festivalPanel : JPanel
- relicPanel : JPanel
+ ContentPanel(contentCardLayout : CardLayout)
+ createHumanContentPanel() : JPanel
+ createKingContentPanel() : JPanel
+ createPoinsettiaContentPanel() : JPanel
+ createDynastyContentPanel() : JPanel
+ createFestivalContentPanel() : JPanel
+ createReicContentPanel() : JPanel
+ createEventContentPanel() : JPanel
ButtonsPanel
- containerPanelCardLayout : CardLayout
- containerPanel : JPanel
- currentPage : int
- totalButtons : int
- cardButtons : RoundedButton[]
- previousButton : RoundedButton
- nextButton : RoundedButton
- backButton : RoundedButton
+ ButtonsPanel (containerPanelCardLayout : CardLayout, containerPanel : JPanel) : void
+ updateCardButtons() : void
+ updatePanel(jPanel : JPanel, btnNext : RoundedButton, btnPrevious : RoundedButton, roundedButtons : RoundedButton[], myCurrentPage : int) : void
ContainerPanel
- containerCardLayout : CardLayout
- index : int
- numberOfPage : int
- objects : Object[]
- detailJPanel : JPanel
- relatedObjectJPanel : JPanel
+ ContainerPanel(containerCardLayout : CardLayout, numberOfPage : int, objects : Object[])
+ createPagination(numberOfPage : int, objects : Object[], containerCardLayout : CardLayout) : JPanel

Figure 3: GUI's class diagram

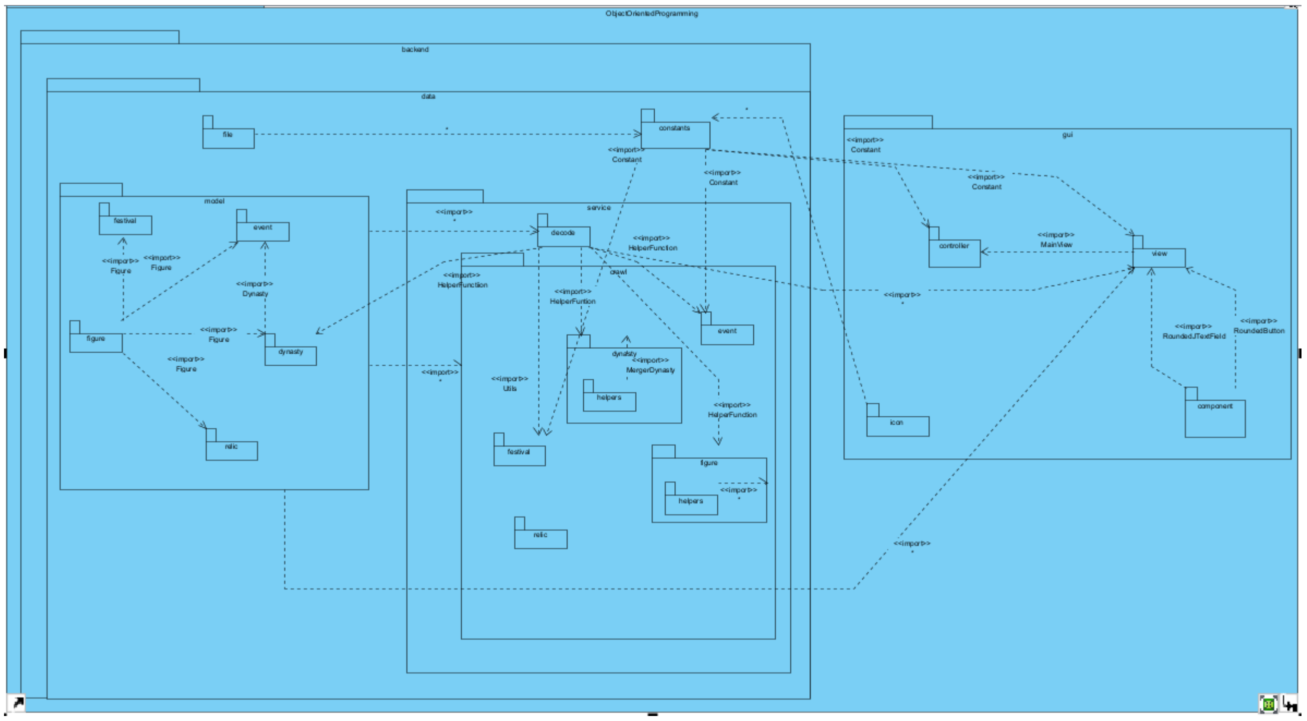


Figure 4: Package diagram

6 Packages features and explanation

To differentiate between the view and the data, the project is seperated in 2 main pacakages: *gui* and *backend*

- Firstly, in *gui* package, we create a "data" subpackage and separate it into 4 sub-packages:
  1. "controller": Handle the action performed by users when they interact with the menu bar
  2. "icon": Contain the icons used in the project
  3. "view": Contain classes to initialize the graphic user interface for the application
  4. "component": Contain classes to initialize other components (buttons, textFields, . . .)
- Secondly, in *backend* package, multiple sub-packages are also separated into:
  1. "constant": Contain constant variables for the project (name, file path, . . .)
  2. "file": Hold the JSON files of the project that are generated from scraping methods in other classes
  3. "model": Contain object classes (Dynasty, Figure, ... Each is placed into another sub-package with the same name
  4. "service": This package contains 2 subpackages: "crawl" and "decode", serving exactly the purposes by the name: Crawling data from the internet" and "decode to array lists for gui display respectively

Specific explanation:

- Handling "data" in backend
  - Use crawling methods to scrape data from multiple websites, then apply algorithms to remove duplication, . . . and save data lists to JSON files

  - Use decoding methods to parse data from JSON format to arraylists following in "model" package

- gui design and application's features
  - Apply Java Swing, create components, views and initialize them "component" and "view" packages
  - Display the decoded data to the UI
  - Create controllers to handle user actions: Browse, search, change pages, ...

7 Object oriented programming techniques

The project initially apply all 4 basic concept of OOP - Object oriented programming: Encapsulation, Polymorphism, Inheritance and Abstract:

1. Encapsulation: By creating classes, we already apply the first technique of OOP. Attributes and methods are encapsulated inside a class and set access-modifier
2. Inheritance
   - We create King and Poinsettia class inherit from Figure with extra attributes
   - Event inherits EventInit
   - Relic inherits Type, Type inherits Province
3. Abstract
   - Interface IObject: contains 5 unimplemented methods Class Figure, Dynasty, EventInit, Relic, Festival implements IObject
4. Polymorphism:
   - Inside the figure list that is shown to the UI there are also Kings and Figures inside which can be downcast to call specific attributes and methods.
   - List EventInit has element as Event (we use Upcast technique)
5. Aggregation: Inside almost every object classes, we also include related data
   - In Dynasty class there is a list of related Figures
   - Event class also contains its related Figure and Dynasty
   - Festival class also contains Figure
6. Generic programming
   - When user want to search for any information, "searchObject" will handle searching functionality: an arraylist of any type is passed in and return the corresponding data type
7. Overriding
   - Figure, Dynasty, EventInit, Relic, Festival classes implement the interface so they override unimplement methods of IObject
   - In FigureScraper and DynastyScraper, they both implement Scraper so they override **start** and **merge** methods of Scraper

Using these techniques above we can obtain several benefits:

- Using Inheritance allows code reuse, promotes specialization.
- Encapsulation can improves data security, modularity, and maintainability in code.
- Polimorphism provides object-oriented design for flexibility, extensibility, and modularity.

- Abtraction will simplifies complex systems, reduces code duplication, and promotes efficient collaboration.

Howerver, they also come with some disadvantages:

- Deep inheritance hierarchies and tight coupling may increase complexity and maintainability problems.
- Encapsulation overuse increases complexity, challenges accessing data.
- Polymorphism introduces indirection, performance overhead, requires careful handling.
- Abstract overuse of abstraction can lead to disconnected codebase, challenging comprehension and maintenance.

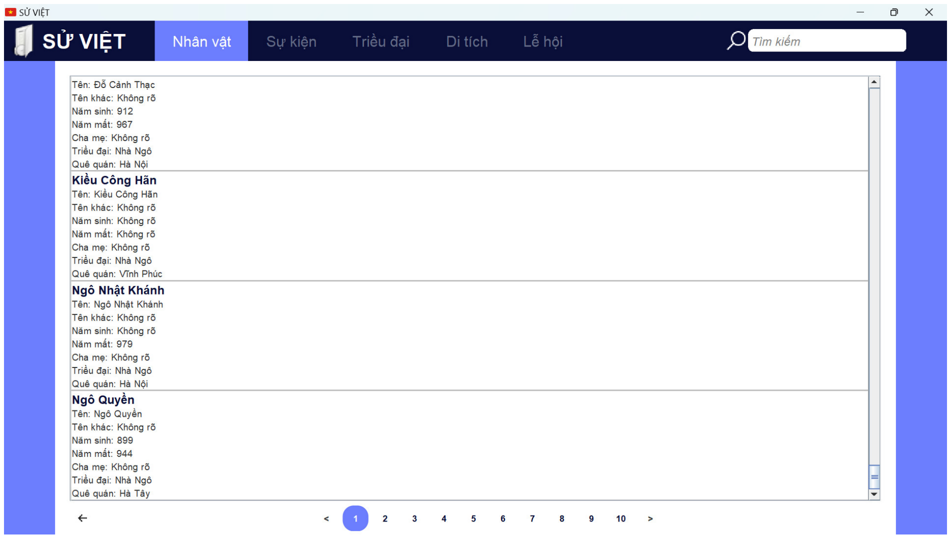
8 Technologies & Algorithms

This project utilize several technologies and algorithms: Java Swing, Jsoup ...

- **WindowBuilder**: creating a graphic user interface using drag-and-drop
- **Java Swing**: A Java framework for building desktop gui applications. It offers a comprehensive set of components and tools for creating interactive user interfaces. With Swing, developers can create platform-independent applications with customizable look and feel.
- **Jsoup**: A Java library that provides a convenient way to parse and manipulate HTML documents. It allows developers to extract data from web pages, modify their contents, and perform web scraping tasks. With JSoup, Java developers can easily handle HTML parsing and interact with web content in their applications.
- **Gson**: A Java library developed by Google that provides an easy way to convert Java objects to JSON representation and vice versa. It simplifies the process of working with JSON data in Java applications by handling the serialization and deserialization of objects automatically. Gson supports various data types, including custom objects, collections, and arrays, making it a powerful tool for working with JSON in Java.
- About algorithms, we apply **Linear Searching**: Sequentially search through the entire array of data, excluding elements that do not match the search keyword in the search field

9 Application demo and features

Here is our link to the project's github repository: GitHub
Upon initial run, we have the home screen as below



Tên: Đỗ Cảnh Thạc
Tên khác: Không rõ
Năm sinh: 912
Năm mất: 967
Cha mẹ: Không rõ
Triều đại: Nhà Ngô
Quê quán: Hà Nội

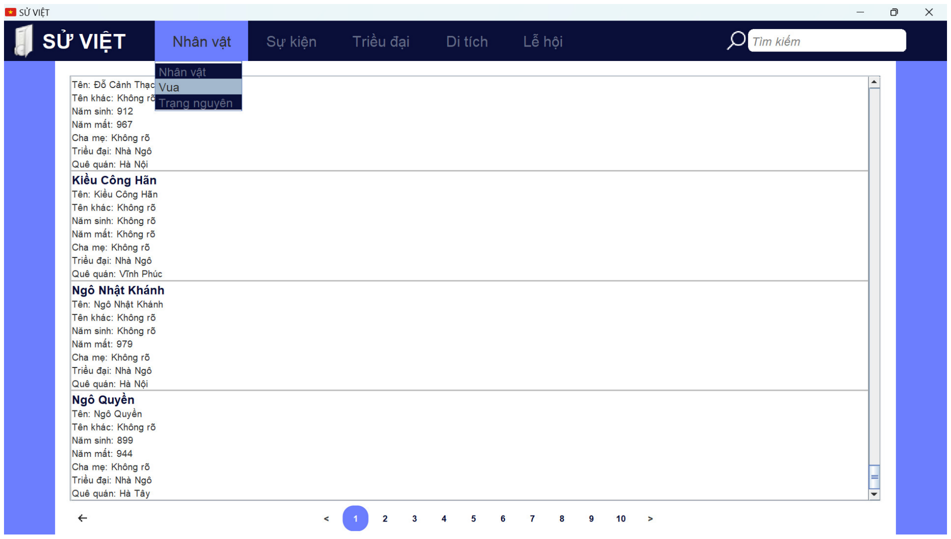
Kiều Công Hãn
Tên: Kiều Công Hãn
Tên khác: Không rõ
Năm sinh: Không rõ
Năm mất: Không rõ
Cha mẹ: Không rõ
Triều đại: Nhà Ngô
Quê quán: Vĩnh Phúc

Ngô Nhật Khánh
Tên: Ngô Nhật Khánh
Tên khác: Không rõ
Năm sinh: Không rõ
Năm mất: 979
Cha mẹ: Không rõ
Triều đại: Nhà Ngô
Quê quán: Hà Nội

Ngô Quyền
Tên: Ngô Quyền
Tên khác: Không rõ
Năm sinh: 899
Năm mất: 944
Cha mẹ: Không rõ
Triều đại: Nhà Ngô
Quê quán: Hà Tây

Figure 5: Initial setup

For user to select the information desired, click on any section on Menu bar to open a drop-down. From there, user select the tab they want to see



Tên: Đỗ Cảnh Thạc
Tên khác: Không rõ
Năm sinh: 912
Năm mất: 967
Cha mẹ: Không rõ
Triều đại: Nhà Ngô
Quê quán: Hà Nội

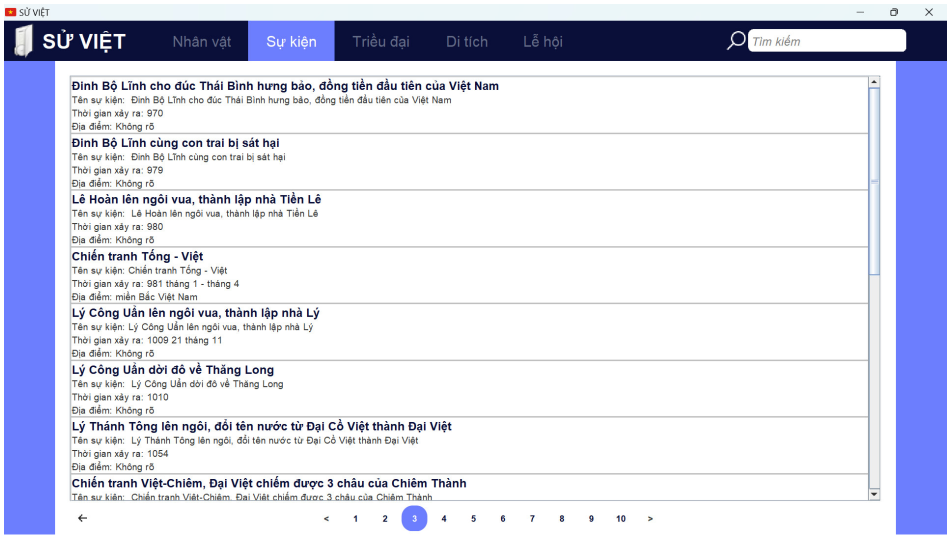
Kiều Công Hãn
Tên: Kiều Công Hãn
Tên khác: Không rõ
Năm sinh: Không rõ
Năm mất: Không rõ
Cha mẹ: Không rõ
Triều đại: Nhà Ngô
Quê quán: Vĩnh Phúc

Ngô Nhật Khánh
Tên: Ngô Nhật Khánh
Tên khác: Không rõ
Năm sinh: Không rõ
Năm mất: 979
Cha mẹ: Không rõ
Triều đại: Nhà Ngô
Quê quán: Hà Nội

Ngô Quyền
Tên: Ngô Quyền
Tên khác: Không rõ
Năm sinh: 899
Năm mất: 944
Cha mẹ: Không rõ
Triều đại: Nhà Ngô
Quê quán: Hà Tây

Figure 6: Dropdown menu

On user selection, the interface will display brief information about related data. User can select pages to be able to view more if there's any



Đinh Bộ Lĩnh cho đúc Thái Bình hưng bảo, đồng tiền đầu tiên của Việt Nam
Tên sự kiện: Đinh Bộ Lĩnh cho đúc Thái Bình hưng bảo, đồng tiền đầu tiên của Việt Nam
Thời gian xảy ra: 970
Địa điểm: Không rõ

Đinh Bộ Lĩnh cùng con trai bị sát hại
Tên sự kiện: Đinh Bộ Lĩnh cùng con trai bị sát hại
Thời gian xảy ra: 979
Địa điểm: Không rõ

Lê Hoàn lên ngôi vua, thành lập nhà Tiền Lê
Tên sự kiện: Lê Hoàn lên ngôi vua, thành lập nhà Tiền Lê
Thời gian xảy ra: 980
Địa điểm: Không rõ

Chiến tranh Tống - Việt
Tên sự kiện: Chiến tranh Tống - Việt
Thời gian xảy ra: 981 tháng 1 - tháng 4
Địa điểm: miền Bắc Việt Nam

Lý Công Uẩn lên ngôi vua, thành lập nhà Lý
Tên sự kiện: Lý Công Uẩn lên ngôi vua, thành lập nhà Lý
Thời gian xảy ra: 1009 21 tháng 11
Địa điểm: Không rõ

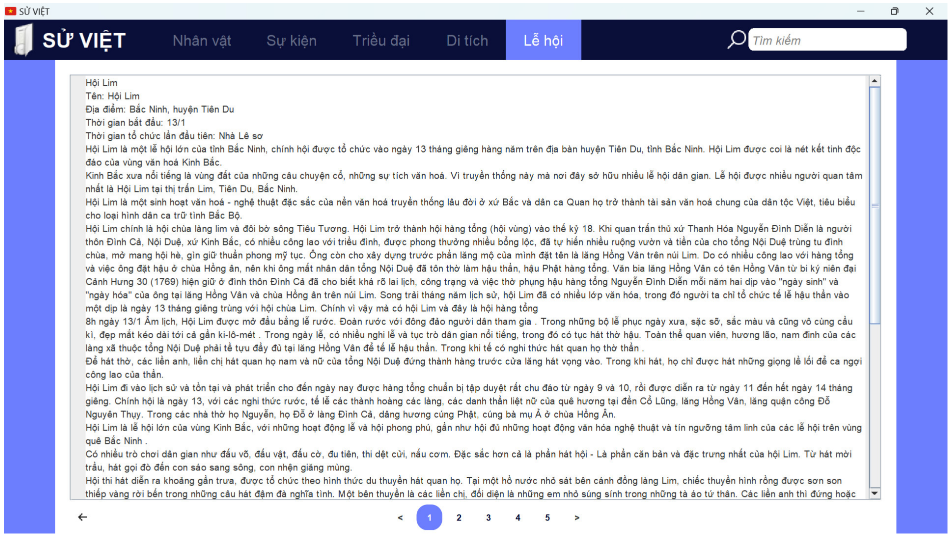
Lý Công Uẩn dời đô về Thăng Long
Tên sự kiện: Lý Công Uẩn dời đô về Thăng Long
Thời gian xảy ra: 1010
Địa điểm: Không rõ

Lý Thánh Tông lên ngôi, đổi tên nước từ Đại Cồ Việt thành Đại Việt
Tên sự kiện: Lý Thánh Tông lên ngôi, đổi tên nước từ Đại Cồ Việt thành Đại Việt
Thời gian xảy ra: 1054
Địa điểm: Không rõ

Chiến tranh Việt-Chiêm, Đại Việt chiếm được 3 châu của Chiêm Thành
Tên sự kiện: Chiến tranh Việt-Chiêm, Đại Việt chiếm được 3 châu của Chiêm Thành

Figure 7: Brief search result information

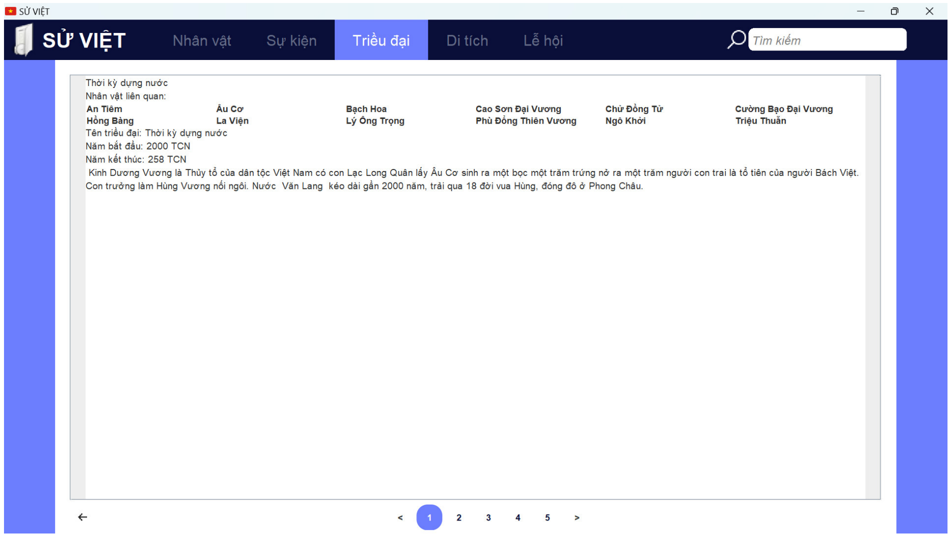
From there, to view detail of a data object, simply click on the name of the data to move to a new page



Hội Lim
Tên: Hội Lim
Địa điểm: Bắc Ninh, huyện Tiên Du
Thời gian bắt đầu: 13/1
Thời gian tổ chức lần đầu tiên: Nhà Lê sơ
Hội Lim là một lễ hội lớn của tỉnh Bắc Ninh, chính hội được tổ chức vào ngày 13 tháng giêng hàng năm trên địa bàn huyện Tiên Du, tỉnh Bắc Ninh. Hội Lim được coi là nét kết tinh độc đáo của vùng văn hoá Kinh Bắc.
Kinh Bắc xưa nổi tiếng là vùng đất của những câu chuyện cổ, những sự tích văn hoá. Vì truyền thống này mà nơi đây sở hữu nhiều lễ hội dân gian. Lễ hội được nhiều người quan tâm nhất là Hội Lim tại thị trấn Lim, Tiên Du, Bắc Ninh.
Hội Lim là một sinh hoạt văn hoá - nghệ thuật đặc sắc của nền văn hoá truyền thống lâu đời ở xứ Bắc và dân ca Quan họ trở thành tài sản văn hoá chung của dân tộc Việt, tiêu biểu cho loại hình dân ca trữ tình Bắc Bộ.
Hội Lim chính là hội chùa làng lim và đôi bờ sông Tiêu Tương. Hội Lim trở thành hội hàng tổng (hội vùng) vào thế kỷ 18. Khi quan trấn thủ xứ Thanh Hóa Nguyễn Đình Diễn là người thôn Đình Cả, Nội Duệ, xứ Kinh Bắc, có nhiều công lao với triều đình, được phong thưởng nhiều bổng lộc, đã tự hiến nhiều ruộng vườn và tiền của cho tổng Nội Duệ trùng tu đình chùa, mở mang hội hè, gìn giữ thuần phong mỹ tục. Ông còn cho xây dựng trước phần lăng mộ của mình đặt tên là lăng Hồng Vân trên núi Lim. Do có nhiều công lao với hàng tổng và việc ông đặt hậu ở chùa Hồng ân, nên khi ông mất nhân dân tổng Nội Duệ đã tôn thờ làm hậu thần, hậu Phật hàng tổng. Văn bia lăng Hồng Vân có tên Hồng Vân từ bi ký niên đại Cảnh Hưng 30 (1769) hiện giữ ở đình thôn Đình Cả đã cho biết khá rõ lai lịch, công trạng và việc thờ phụng hậu hàng tổng Nguyễn Đình Diễn mỗi năm hai dịp vào "ngày sinh" và "ngày hóa" của ông tại lăng Hồng Vân và chùa Hồng ân trên núi Lim. Song trải tháng năm lịch sử, hội Lim đã có nhiều lớp văn hóa, trong đó người ta chỉ tổ chức tế lễ hậu thần vào một dịp là ngày 13 tháng giêng trùng với hội chùa Lim. Chính vì vậy mà có hội Lim và đây là hội hàng tổng
8h ngày 13/1 Âm lịch, Hội Lim được mở đầu bằng lễ rước. Đoàn rước với đông đảo người dân tham gia . Trong những bộ lễ phục ngày xưa, sặc sỡ, sắc màu và cũng vô cùng cầu kì, đẹp mắt kéo dài tới cả gần ki-lô-mét . Trong ngày lễ, có nhiều nghi lễ và tục trò dân gian nổi tiếng, trong đó có tục hát thờ hậu. Toàn thể quan viên, hương lão, nam đinh của các làng xã thuộc tổng Nội Duệ phải tề tựu đầy đủ tại lăng Hồng Vân để tế lễ hậu thần. Trong khi tế có nghi thức hát quan họ thờ thần .
Để hát thờ, các liền anh, liền chị hát quan họ nam và nữ của tổng Nội Duệ đứng thành hàng trước cửa lăng hát vọng vào. Trong khi hát, họ chỉ được hát những giọng lề lối để ca ngợi công lao của thần.
Hội Lim đi vào lịch sử và tồn tại và phát triển cho đến ngày nay được hàng tổng chuẩn bị tập duyệt rất chu đáo từ ngày 9 và 10, rồi được diễn ra từ ngày 11 đến hết ngày 14 tháng giêng. Chính hội là ngày 13, với các nghi thức rước, tế lễ các thành hoàng các làng, các danh thần liệt nữ của quê hương tại đền Cổ Lũng, lăng Hồng Vân, lăng quận công Đỗ Nguyên Thụy. Trong các nhà thờ họ Nguyễn, họ Đỗ ở làng Đình Cả, dâng hương cúng Phật, cúng bà mụ Ả ở chùa Hồng Ân.
Hội Lim là lễ hội lớn của vùng Kinh Bắc, với những hoạt động lễ và hội phong phú, gần như hội đủ những hoạt động văn hóa nghệ thuật và tín ngưỡng tâm linh của các lễ hội trên vùng quê Bắc Ninh .
Có nhiều trò chơi dân gian như đấu võ, đấu vật, đấu cờ, đu tiên, thi dệt cửi, nấu cơm. Đặc sắc hơn cả là phần hát hội - Là phần căn bản và đặc trưng nhất của hội Lim. Từ hát mời trầu, hát gọi đò đến con sáo sang sông, con nhện giăng mùng.
Hội thi hát diễn ra khoảng gần trưa, được tổ chức theo hình thức du thuyền hát quan họ. Tại một hồ nước nhỏ sát bên cánh đồng làng Lim, chiếc thuyền hình rồng được sơn son thiếp vàng rời bến trong những câu hát đậm đà nghĩa tình. Một bên thuyền là các liền chị, đối diện là những em nhỏ súng sính trong những tà áo tứ thân. Các liền anh thì đứng hoặc

Figure 8: Detail screen

Related data (if exists) will also be displayed inside detail view. User can now click on the data to move to its own detail page



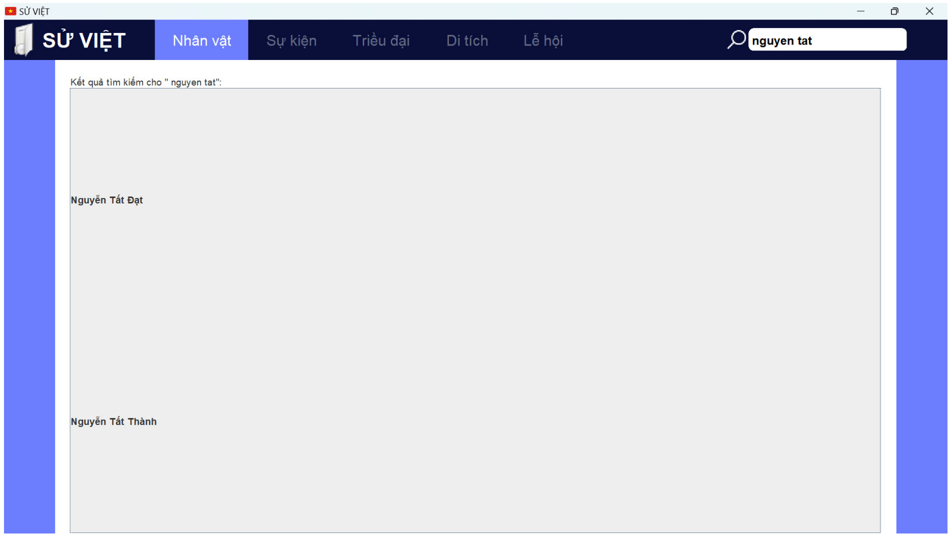
Thời kỳ dựng nước
Nhân vật liên quan:

An Tiêm	**Âu Cơ**	**Bạch Hoa**	**Cao Sơn Đại Vương**	**Chử Đồng Tử**	**Cường Bạo Đại Vương**
Hồng Bàng	**La Viện**	**Lý Ông Trọng**	**Phù Đổng Thiên Vương**	**Ngô Khởi**	**Triệu Thuẫn**

Tên triều đại: Thời kỳ dựng nước
Năm bắt đầu: 2000 TCN
Năm kết thúc: 258 TCN
Kinh Dương Vương là Thủy tổ của dân tộc Việt Nam có con Lạc Long Quân lấy Âu Cơ sinh ra một bọc một trăm trứng nở ra một trăm người con trai là tổ tiên của người Bách Việt. Con trưởng làm Hùng Vương nối ngôi. Nước Văn Lang kéo dài gần 2000 năm, trải qua 18 đời vua Hùng, đóng đô ở Phong Châu.

Figure 9: Related data

Finally, to search a desired data, user can go to that page and enter the name to the search box. Search functions will search the list by keywords and id to return all related to the screen



Kết quả tìm kiếm cho " nguyen tat":

Nguyễn Tất Đạt

Nguyễn Tất Thành

Figure 10: Search function

10 Conclusion

- **Advantages:**
  - Our project has successfully developed a historical information application using Java, incorporating various object-oriented programming (OOP) techniques and technologies that we have learned during the course.
  - The application provides a comprehensive and interactive platform for users to explore and learn about Vietnam's history.
- **Disadvantages:**
  - Although we scraped data from multiple websites, there may still be limitations in terms of the available data.
  - While we made efforts to remove duplicates during the data scraping process, there is still a possibility of encountering duplicate or inconsistent data in the dataset.