NEWS AGGREGATOR APPLICATION DOCUMENTATION

Object-oriented programming

Lecturer: Mr. Minh Vu Thanh
Group 31: Anh Em Cot Cheo

Student 1 Name: Nguyen Giang Huy
Student 2 Name: Hua Nam Huy
Student 3 Name: Nguyen Phan Duc Anh
Student 4 Name: Tran Hoang Vu

Student 1 Number: s3836454
Student 2 Number: s3881103
Student 4 Number: s3915181
Student 3 Number: s3915185

RMIT UNIVERSITY

Product description

Our group set out to design an application to scape and display news articles from numerous of existing news outlets (e.g. VnExpress, ZingNews, Tuoi Tre) by using JavaFX and a touch of CSS. Our application is capable of pulling all of the latest articles from listed online sources and organizes them in sub-categories namely Newest, Covid, Politics, Business, Technology, Health, Sports, Entertainment, Sports, World, Others.

Aiming to maximize user's experience, we've designed a user-friendly navigation system allowing smooth first-time visits.

Running instructions

- Step 1: Install and set up IntelliJ IDEA.
- Step 2: Install and unzip openjdk-18
- Step 3: Open IntelliJ, click on "Get From Version Control", then copy past the Github link: s3915185/OOP_FINALGROUPPROJECT (github.com) into URL box.
- Step 4: Select File -> Project Structure. A new window will be displayed, "Project SDK", click on arrow -> Add SDK -> JDK -> choose JDK 18 (unzipped earlier).
- Step 5: Click on "Build Project" which helps IntelliJ to detect the new JDK.
  Step 6: Go to src -> sample -> main and run program.

Features included

News categories

We've designed a horizontal menu bar for simple navigation, there are a total 10 categories namely Newest, Covid, Politics, Business, Technology, Health, Sports, Entertainment, World, Others. Users can intuitively choose news category they want to read and this bar is visible all the time for moving around the application.

Figure 1. Application Menu Bar

Each category will have 50 news articles scraped from 5 news outlets below:

1. VnExpress(Links to an external site.) (Links to an external site.)
2. ZingNews(Links to an external site.) (Links to an external site.)
3. TuoiTre(Links to an external site.) (Links to an external site.)
4. ThanhNien(Links to an external site.) (Links to an external site.)
5. NhanDan(Links to an external site.)

When clicked on one of the category, a function that directs to that page is written in a base controller class.

```
public void toNewPage(ActionEvent actionEvent) throws IOException {
    FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "/FXMLs/NewestPage.fxml"));
    base = loader.load();
    NewestController newestController = loader.getController();
    newestController.setListPage(0);
    stage = (Stage) ((Node)actionEvent.getSource()).getScene().getWindow();
    stage.getScene().setRoot(base);
    stage.show();
}
```

Figure 2. Sample of ActionEvent.

Loading Screen

When the application is started, a loading screen pops up to let users know the app is currently working. We used a live animation of a rolling circle and eye-capturing background for better user's engagement.

Figure 3. Loading Screen

Article preview

After the news is stored. It will be distributed into 5 pages (10 news each page). The content will be displayed through FXML elements using set methods. For each set method, the method will take an array list of FXML elements (labels or image views), an integer, and the list of news in the controller as parameters. The method will create the loop, where it will take the news' attributes, starting from the integer (parameter), and set to the FXML elements.

Figure 4. Slot content previews

Article preview includes the headline, a short snippet of the actual article, time when it was published and its source.

```
public void setDescrips(ArrayList<Label> labelList, int begin, NewsLoader newsList){
    int count = begin;
    for (Label description: labelList) {
        description.setFont(Font.font( family: "Time New Roman", FontWeight.NORMAL,  size: 15));
        description.setAlignment(Pos.TOP_LEFT);
        description.setWrapText(true);
        description.setText(newsList.getNews(count).getDescrip());
        count++;
    }
}
```

Figure 5. Example of a set method.

Article contents

Article is opened with a different layout than the homepage. Its design includes an option to return to previous page and still has a menu bar on the top of the page.

Figure 6. Layout preview.

Web Scraper

Our implementation scrapes data from various news outlets display on both category page and article contents. Thus, the external tool which is applied to the systems to get the content is Jsoup. However, each news outlet has a different HTML structure, so the system needs to be divided into five different scraping methods for five outlets. In each scraping method, there would be five different algorithms to access the HTML content of each outlet into a news list as a class object. To separate each category link from the other, the system has 10 classes represent 10 categories of the article.

```
public void loadZingNews(String url) throws IOException {
    String originalURL = "https://zingnews.vn";
    String newsURL;
    String title;
    String descrip;
    String imageURL;

    Document document = Jsoup.connect(url).get();
    Elements article = document.select( cssQuery: "article.article-item");
    String[] scripts = article.toString().split( regex: "</article>");
    for (String script : scripts) {...}
    }
```

Figure 7. One of the functions to scrape data.

Chronological article display

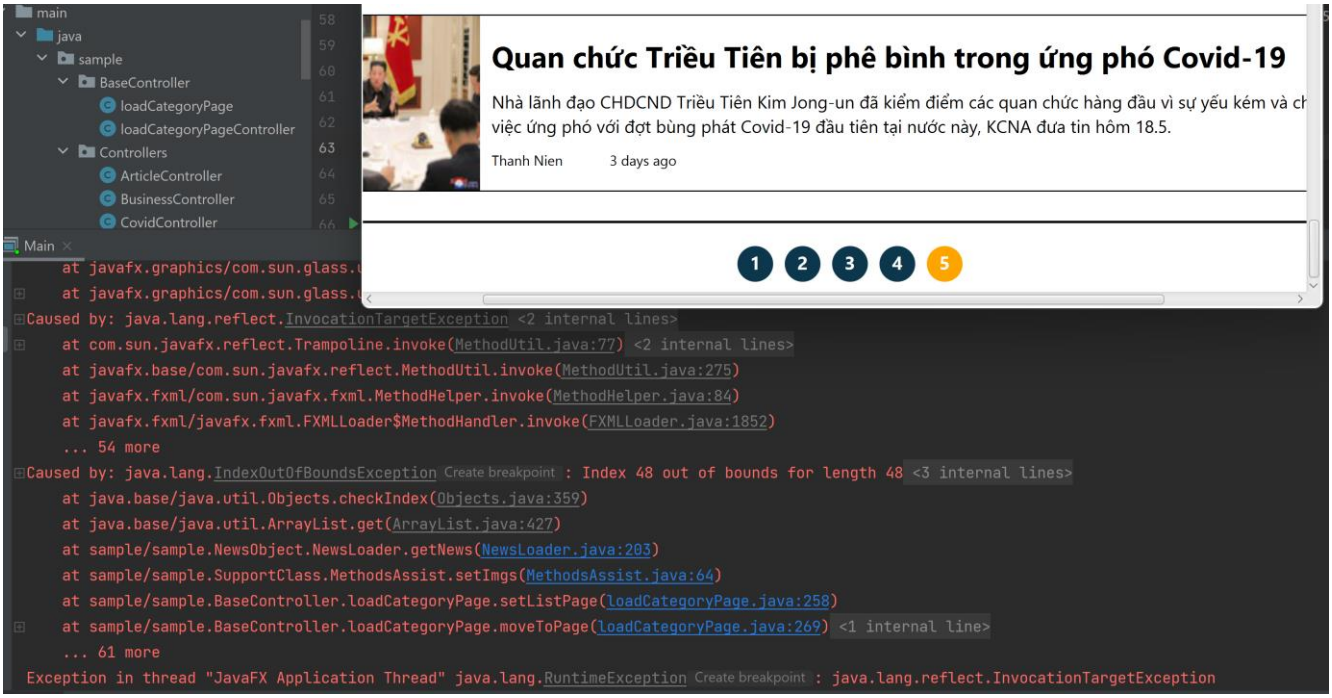
After scraping data from the web, the application displays article in chronological order to from latest to oldest with time stamp in slot content preview.

```java
private void news_sort() {
    // count for 50 articles with array lists to save dats
    int counter = 0;
    ArrayList<Time> releasedDates = new ArrayList<>();



    // This is to modify the time in the list of news
    for (int i = 0; i < this.newsLoader.getSize(); i++) {
        try {
            // go to link of that article
            Document newsDocument = Jsoup.connect(newsLoader.getNews(i).getNewsURL()).timeout(200).get();

            // get the link of article
            String timeFromSource = newsDocument.select( cssQuery: "meta[itemprop=datePublished]").attr( attributeKey: "content");
            // if empty, then have the another way to get the value - this is for nhanDan
            if (timeFromSource.isEmpty()) {
if (timeFromSource.isEmpty()) {
    timeFromSource = newsDocument.select( cssQuery: "div.box-date").text();
    if (!timeFromSource.isEmpty()) {
        String date = timeFromSource.split( regex: ", ")[1];
        String time = timeFromSource.split( regex: ", ")[2];
        String day = date.split( regex: "-")[0];
        String month = date.split( regex: "-")[1];
        String year = date.split( regex: "-")[2];
        String hour = time.split( regex: ":")[0];
        String minutes = time.split( regex: ":")[1];
        timeFromSource = year + "-" + month + "-" + day + "T" + hour + ":" + minutes + ":00+07:00";
    }
}


// This is for other webtsites
if (timeFromSource.isEmpty()) {
    timeFromSource = newsDocument.select( cssQuery: "meta[property=article:published_time]").attr( attributeKey: "content");
}
```

Figure 8. Example of chronological article display.

Known bugs & debug

Bug: Our implementation is sometimes unable to scrape 50 articles and therefore terminal throws exceptions Index out of bound.



Debug: After trying to adjust the time function and newsloader functions, we realized that this may be contributable to internet connection. Debugging suggestion might be to get a strong and stable internet connection.

Bug: Uneven distribution of articles from given news outlet.
Debug: Our application can have more articles from one news outlet than another. We do not actually have a solution for this since Jsoup pulls data randomly from URLs given.

Bug: Occasional article duplication.
Debug: This is likely due to error from the web page that we tried to scrape because if this was in fact a bug in our code, there would be a duplication from every news article we scrape but this is not the case in our implementation.

Bug: The author of the articles in Zingnews is encrypted or locked information, thereby cannot use Jsoup to scrape.
Debug: There will be a preview slot content without brief description, but it still has a headline and published time.

Bug: Data scraping algorithms occasionally pull articles from long time ago from NhanDan.
Debug: None

Project demo video

Link: https://www.youtube.com/watch?v=bB_JhVxQTsw

Hua Nam Huy: (0:00 – 1:45) Present UI design.
Nguyen Phan Duc Anh: (1:46 – 3:16) UML diagram.

Tran Hoang Vu: (3:17 – 4:51) Data scraping algorithms.
Nguyen Giang Huy: (4:52 – 6:15) Bug and workarounds.

Acknowledgement

1. Jsoup, “Jsoup: Java HTML parser”, Jsoup, 2021. [Online]: Available: https://jsoup.org. [Accessed: 19 May-2022].
2. JavaFX Animations: https://www.youtube.com/watch?v=UdGiuDDi7Rg
3. JavaFX Animations and Transition Splash Screen / Welcome Screen - Netbeans and SceneBuilder: https://www.youtube.com/watch?v=Fy0ZVF7EPC4
4. JavaFX Splash Screen 2 : Loading In a Seperate Window: https://www.youtube.com/watch?v=f06uUtkmtDE
5. HTML & CSS Full Course 🌎-Learn to create a website-【***Free***】: https://www.youtube.com/watch?v=cyuzt1Dp8X8
6. JavaFx Event Handling Information: https://www.tutorialspoint.com/javafx/javafx_event_handling.htm
7. HTML Parsing Jsoup: https://www.scrapingbee.com/blog/java-parse-html-jsoup/
8. Selection Sort: https://www.programiz.com/dsa/selection-sort
9. Scene Switching: https://www.youtube.com/watch?v=hcM-R-YOKkQ
10. JavaFX and Threads: https://www.youtube.com/watch?v=Xb6j8VfHxJo
11. 5Java Web Scraping: https://www.youtube.com/watch?v=yw7B85174JQ

References

[1] M. V. Thanh, "Assessment Details Documentation," [Online]. Available: https://rmit.instructure.com/courses/101186/assignments/676518.

[2] M. V. Thanh, "Lecture Slides," [Online]. Available: https://rmit.instructure.com/courses/101186/modules.