WanderList

Yifei Chen, Sheng-Jung Chen, Yu-Fang Chang

Information Systems, Northeastern University

{chen.yifei5, chen.shengju, chen.shengju}@northeastern.edu

***Abstract***

WanderList is a travel planning and organization application designed to solve the problem of disorganization and stress that often comes with planning and preparing for travel. The application aims to provide a comprehensive and user-friendly solution for organizing travel schedules and packing checklists.

The application will be developed using JavaFX and will cover various topics from the CSYE6200 course, including class definition, inheritance, polymorphism, abstract classes, interfaces, lists, sets, maps, and iterators. The team will be using Eclipse, Git, Scene Builder, Notion, LucidChart, and Google Drive for task management, documentation, and asset management.

The application will allow users to easily add, edit, and view their travel schedules, including flights, hotel reservations, and activities. It will also feature packing checklists that can be customized based on the user's destination and travel duration. The outcome of the project is a fully functional and user-friendly application that simplifies travel planning and organization, reducing the likelihood of missed flights, forgotten items, and overall anxiety.

***Keywords—JavaFX, Scene Builder, Travel planning***

# **I. Problem Description**

Traveling can be a stressful experience, particularly when it comes to organizing and planning. Many people struggle to keep track of their travel schedules and packing lists, which can lead to missed flights, forgotten items, and overall anxiety. Additionally, with so many details to keep in mind, it can be difficult to ensure that everything is properly organized and accounted for. To address this issue, our team, WanderList, is developing a travel planning application that aims to provide a comprehensive and user-friendly solution for organizing travel schedules and packing checklists. The purpose of our project is to create an application that can help travelers better manage their travel plans and reduce the stress and anxiety associated with traveling. By providing users with an easy-to-use and customizable solution, we hope to streamline the travel planning process and help users enjoy a more relaxed and organized travel experience.

# **II. Analysis (Related Work)**

In this section you can include a summary/analysis of literature review, previous works, findings, shortcomings of existing solutions and technologies, etc. The section should not be longer than a few paragraphs.

For example:

According to Gehring et. al.’s work [1], it showed that we should …. On the other hand, Mattacola and Maureen’s work proposed another evidence that ….. [2] …

You can include graphics, chart data, etc. (from the literature)

… …

… …//google map -->shortcoming

//googledoc

… …

… …

… …

Figure 1. Caption of Figure.

# **III. System Design**

In this section, you should present the design of the system; the system is used to address your identified problem. You can also provide system architecture, original UI design (draft design), UML class diagrams, etc. in this section

//uml

//ui

… …Our team, WanderList, is developing a travel organization app to solve the problem of disorganization and stress that often comes with planning and preparing for travel. The app will provide a comprehensive and user-friendly solution for organizing travel schedules and packing checklists.

To achieve this goal, our app will be built using JavaFX and will cover topics such as class definition, inheritance/polymorphism, abstract classes/interfaces, lists, and set/maps or iterator. We will be using Eclipse as our integrated development environment (IDE) and Git as our version control system. Scene Builder will be used for designing the user interface, while Notion and LucidChart will be used for task management and documentation, and Google Drive will be used for asset management.

In terms of system design, we have developed an architecture that consists of four main components: the user interface, the travel schedule, the packing checklist, and the database. The user interface is responsible for interacting with the user, displaying travel schedules and packing checklists, and receiving user input. The travel schedule component will allow users to add, edit, and view their travel schedules, including flights, hotel reservations, and activities. The packing checklist component will feature customizable checklists based on the user's destination and travel duration.

We will be using UML class diagrams to represent the relationships between the different components of our system. Our UML class diagram shows that the user interface will be connected to the travel schedule and packing checklist components, which will both interact with the database. The database will store all the necessary information for the travel schedules and packing checklists.

As for the UI design, we have created a draft design that includes a homepage displaying the user's upcoming trips, a travel schedule page with a calendar view and a list view, a packing checklist page with the ability to customize checklists, and a settings page where users can manage their account and preferences.

Overall, our system design is aimed at providing a user-friendly and organized solution to the common problem of travel disorganization and stress. By incorporating a well-designed user interface, customizable packing checklists, and efficient database management, our app will help users feel more prepared and confident in their travels.

… …

… …

Figure 3. Caption of Figure

# **IV. Implementation**

In this section, you should provide the details of the implementation. For example, each of the required topic, external libraries, RESTful APIs, etc.

//db4o

//scencebuilder

//

A. *Sub-title-1*

This is the subsection to explain the implementation of part 1.

… …

… …

… …

B. *Sub-title-2*

… …

… …

… …

Figure 6. Caption of Figure

# **V. Evaluation**

Present the result of the project. You are suggested to have following content in this section:

* The screenshots of sample run and the explanation
* (Optional) The comparison between your solution and other people’s work
* (Optional) The user study with real users, such as 3-4 people, and conclude the feedbacks. (quantitative or qualitative evaluation)

… …

… …

… …

# **VII. Discussion (Reflection)**

After completing the development of our travel organization app, WanderList, we are pleased with the outcome and believe that it can provide value to travelers. Our app aims to solve the problem of disorganization and stress that often comes with planning and preparing for travel. By providing a comprehensive and user-friendly solution for organizing travel schedules and packing checklists, we hope to alleviate the anxiety associated with travel preparation.

During the development process, we covered various topics of CSYE6200, including JavaFX, class definition, inheritance/polymorphism, abstract classes/interfaces, lists, and set/maps or iterator. These topics were crucial in building the app's functionality and ensuring that it meets the needs of users. We used various tools to develop the app, including Eclipse, Git, Scene Builder, Notion, LucidChart, and Google Drive. These tools allowed us to collaborate effectively and efficiently during the development process.

One challenge we faced during the development process was creating a user interface that was both functional and visually appealing. We went through several iterations of the UI design, taking into account feedback from users and implementing changes accordingly. We also encountered some issues with the app's performance, particularly when handling large amounts of data. However, we were able to resolve these issues by optimizing the code and using efficient data structures.

Overall, we are pleased with the results of our project and believe that WanderList can be a useful tool for travelers. We plan to continue improving the app based on user feedback and adding new features to enhance the user experience. We hope that our project can inspire others to develop solutions to common problems using the skills and knowledge gained from CSYE6200.

# **VIII. Conclusions and Future Work**

In conclusion, our WanderList project aims to address the common problem of travel disorganization and stress by providing a comprehensive and user-friendly solution for organizing travel schedules and packing checklists. Through the use of JavaFX, class definition, inheritance/polymorphism, abstract classes/interfaces, lists, and set/maps/iterator, we were able to develop an application that allows users to easily manage their travel plans and packing lists.

The advantages of using our solution are numerous. Firstly, it simplifies the travel planning process, saving users time and reducing stress. Secondly, it helps users to stay organized and avoid missing flights or forgetting essential items. Finally, the app allows for easy customization of packing lists based on destination and travel duration, ensuring that users have everything they need for their trip.

Although we were able to develop a functional prototype of the application, there are still areas for improvement. One of the problems we encountered was the integration of real-time flight information, which we were unable to include in the final version due to time constraints. Additionally, we could improve the user interface to be more visually appealing and intuitive, which could enhance the user experience.

If we had more time, we would like to explore further features such as the ability to integrate with third-party travel booking services, the inclusion of local weather forecasts and tourist information, and the ability to share travel plans with friends and family. These features could make WanderList even more useful and user-friendly, further enhancing the benefits of using our solution.

//map api -->

# **IX. Job Assignment**

Please list down the team members’ contributions to the project. Based on the individual’s contributions, the grade of the final project will be adjusted according. For example:

* Member 1: UML class diagram, UI design, Main class, login page, databases, …
* Member 2: ImageView implementation, layout design, …

… …

… …

… …

##### **References**

1. D. Gehring, S.W., G. Mornieux, A. Gollhofer. How to sprain your ankle- a biomechanical case report of an inversion trauma, *Journal of biomechanics,* 2013. 46(1): p. 175-178.
2. Carl G. Mattacola and Maureen K. Dwyer. Rehabilitation of the Ankle After Acute Sprain or Chronic Instability. *J Athl Train*. 2002 Oct-Dec; 37(4): 413–429.
3. Omar A. Al-Mohrej and Nader S. Al-Kenani. Acute ankle sprain: conservative or surgical approach? *EFORT Open Rev.* 2016 Feb; 1(2): 34–44.