WanderList

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

Information Systems, Northeastern University

chen.yifei5@northeastern.edu, chen.shengju@northeastern.edu, chang.yuf@northeastern.edu

***Abstract***

WanderList is a travel planning and organization application designed to solve the often confusing and stressful problems of travel planning and preparation. The application is designed to provide a comprehensive and user-friendly solution for organizing travel schedules.

The application will be developed using JavaFX and will cover a variety of topics from the CSYE6200 course including class definitions, inheritance, polymorphism, abstract classes, interfaces, lists, collections, maps, and iterators. The team will use Eclipse, Git, Scene Builder, Notion, and LucidChart for task management, documentation, and asset management.

The application will allow users to easily add, edit and view their travel itineraries, facilitating users to keep track of activity scheduling and attraction bookings. The result of this project is a fully functional, user-friendly application that simplifies travel planning and organization, 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 travelers find it challenging to keep track of all the activities and places they want to visit during their trip. Moreover, scheduling these activities can be difficult due to varying operating hours and days. Our Java app offers a solution to these problems by providing a platform for users to create and schedule their travel plans with ease. With features such as the ability to create an account, save wishlist items, and drag-and-drop scheduling, our app simplifies the travel planning process and ensures that users can efficiently manage their itinerary while on the go.

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

In the realm of travel planning apps, there are several existing solutions, such as TripIt, Google Trips, and Roadtrippers. While these apps offer a range of features such as itinerary management and booking capabilities, they can be overwhelming for users who are looking for a simpler, more streamlined planning experience.

Another shortcoming of existing solutions is the lack of integration with other tools and platforms. For example, while Google Maps allows users to save locations, it does not offer the ability to save date and time information alongside these locations. Similarly, while Microsoft Word and Excel can be used to plan a trip, they can result in a messy and disorganized itinerary that requires manual checking for accuracy.

Our app seeks to address these shortcomings by providing a user-friendly, customizable, and integrated travel planning experience. By allowing users to save wishlist items and drag-and-drop schedule these items onto specific days of their trip, our app streamlines the planning process and ensures that users have a clear and organized itinerary.

# **III. System Design**

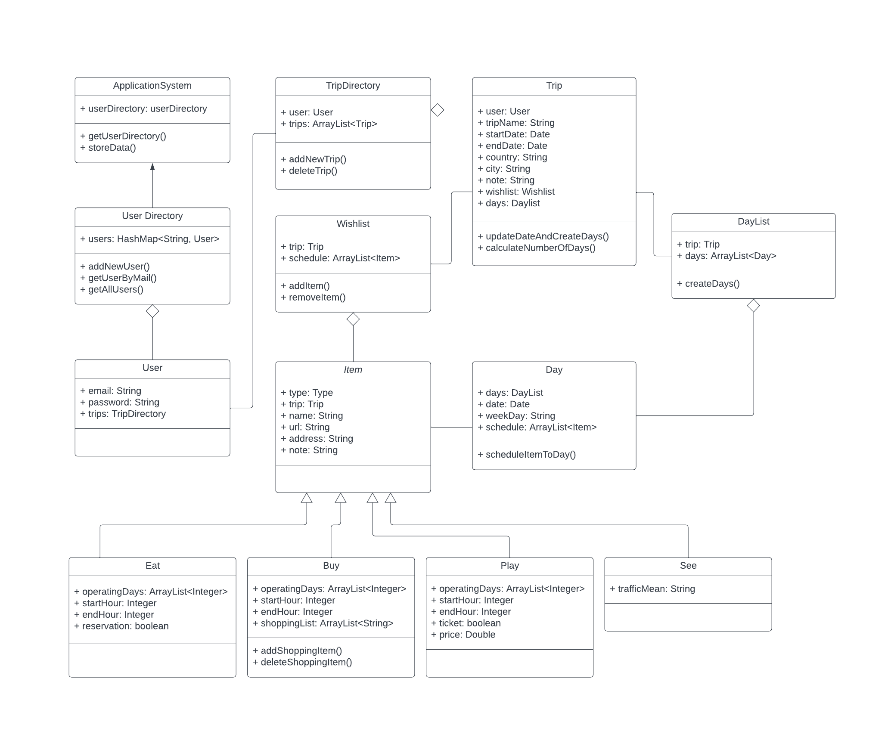
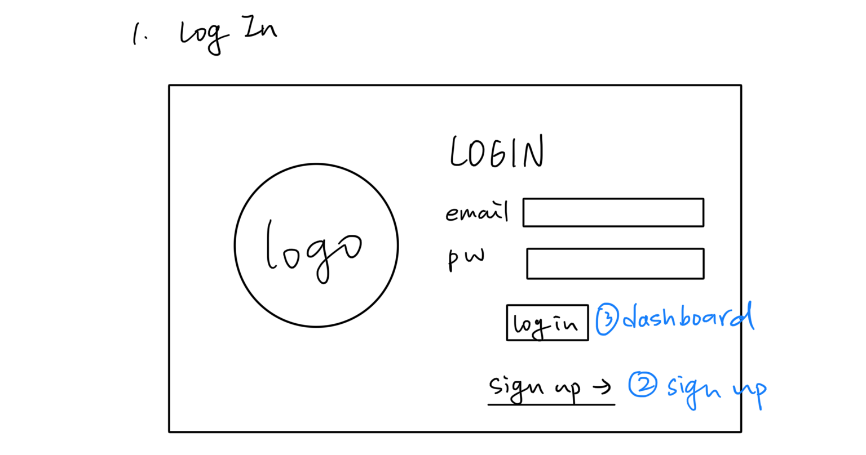


Figure 1. UML diagram



Diagram

Description automatically generated

Diagram

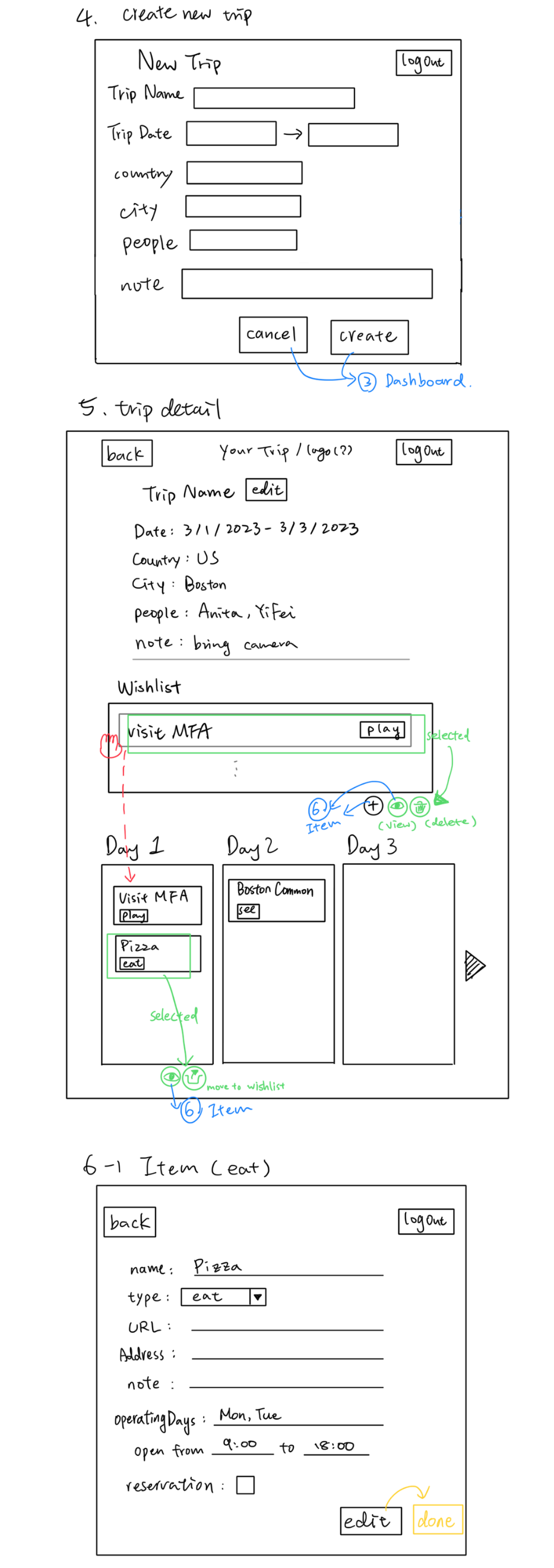
Description automatically generated

Graphical user interface, diagram

Description automatically generated

Graphical user interface, diagram

Description automatically generated



Graphical user interface, application

Description automatically generated

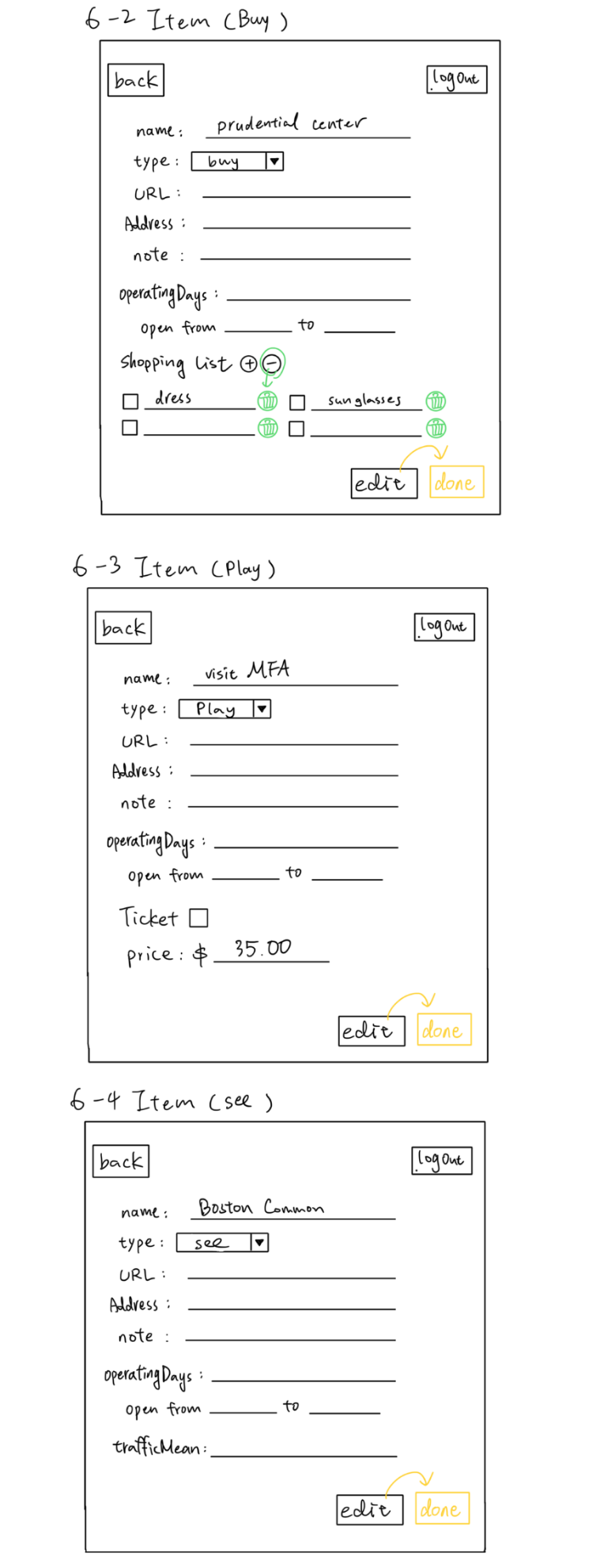


Figure 2. UI design (draft)

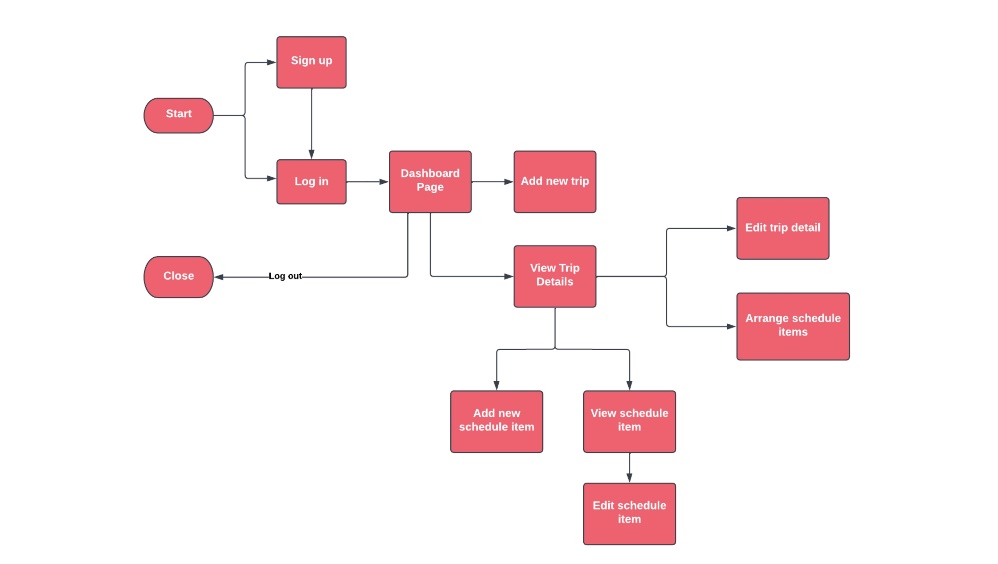


Figure 3. Workflow

# **IV. Implementation**

A. JavaFX

The application is developed with JavaFX, SceneBulider, and FXML for GUI design. We also implemented error handling and user feedback with JavaFX Alert dialogs for various scenarios.

B. Class Definition

Refer to the figure of UML in the previous section: System Design.

C. Map

In UserDirectory class, we store user information in a HashMap, with user email as key and User object as value. In the sign up page, we can validate user input for registration, and interact with UserDirectory to create or check for existing users. In the login page, it can validate user credentials by comparing the input with stored user data.

D. List

ArrayList has been used to store data in higher level objects. TripDirectory’ s data field contains an ArrayList of Trip to store all trip data of the user. Wishlist class contains an ArrayList of Item to store all schedule items. DayList class contains an ArrayList of Day to have hold specific day information and item scheduled.

E. Abstract Class

Item is an abstract class, providing common properties like type, name, url, address, and a set of common methods.

F. Inheritance

Eat, Play, See, and Buy classes extend from the Item abstract class, each class represents a different activity type with specific properties and additional methods.

G. Polymorphism

Methods defined with Item as parameters can take Eat, Play, See, and Buy as arguments.

H. Iterator

DashboardController manages the display of user information and trips. It utilizes an iterator to traverse through the user's trips, splits them into upcoming and past trips with the current date, and displays them in the dashboard.

I. db4o

We utilized db4o as our database solution for this app. It was chosen due to its object-oriented approach and ease of use. We created classes that map to the data that needs to be stored and used db4o to persist and retrieve this data. This allowed us to store and retrieve objects directly, instead of having to use SQL queries to extract the relevant information.

# **V. Evaluation**

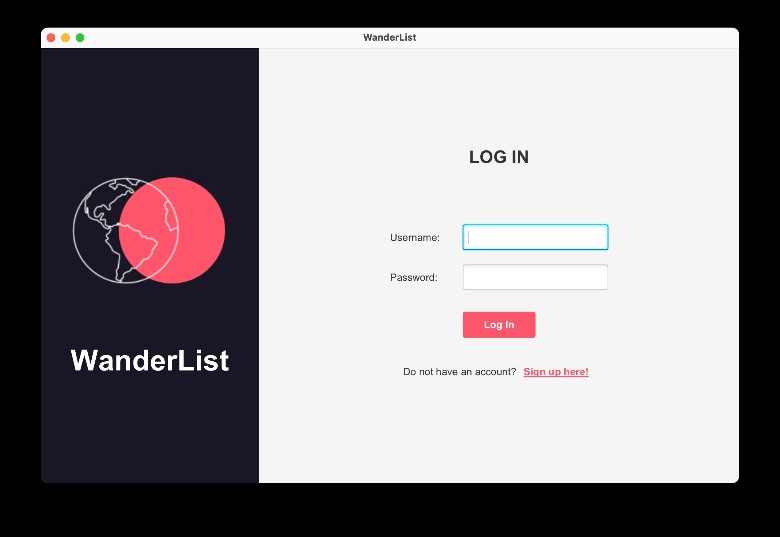


Figure 4. Login page

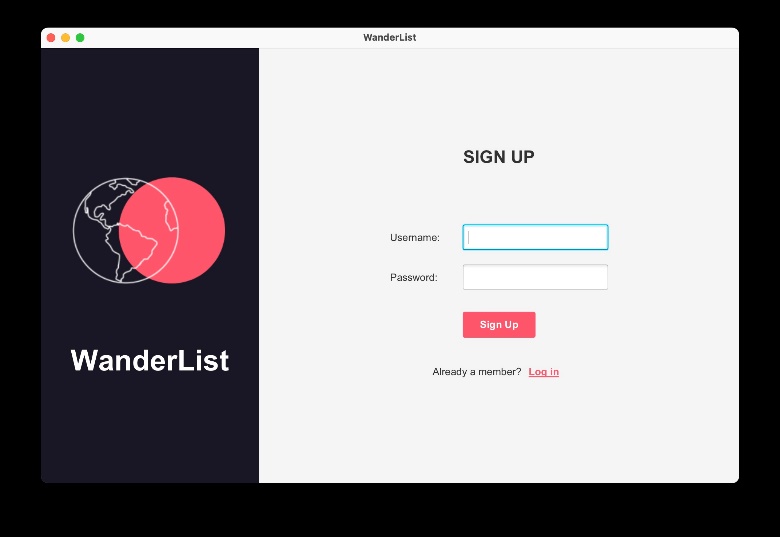


Figure 5. Sign up page

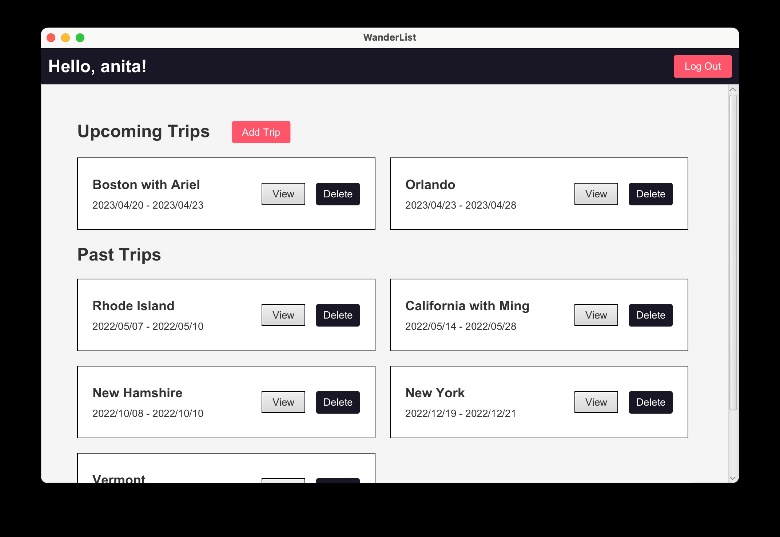


Figure 6. Dashboard page displaying all the user’s trip

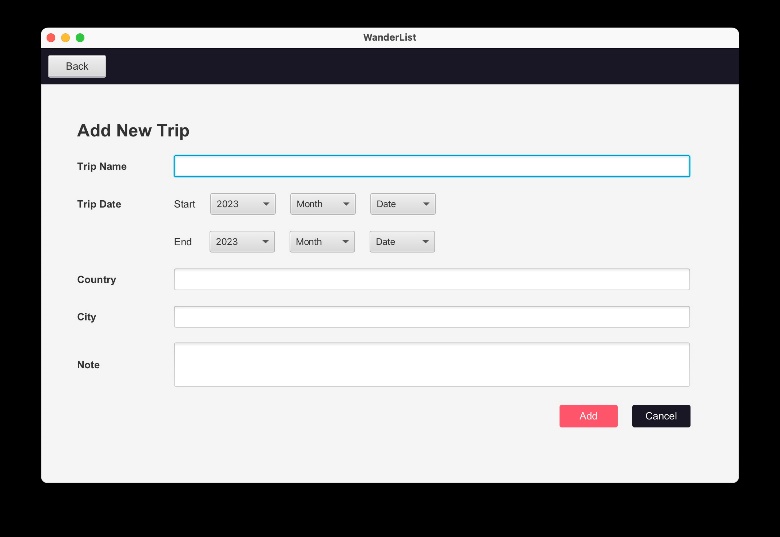


Figure 7. Add new trip age

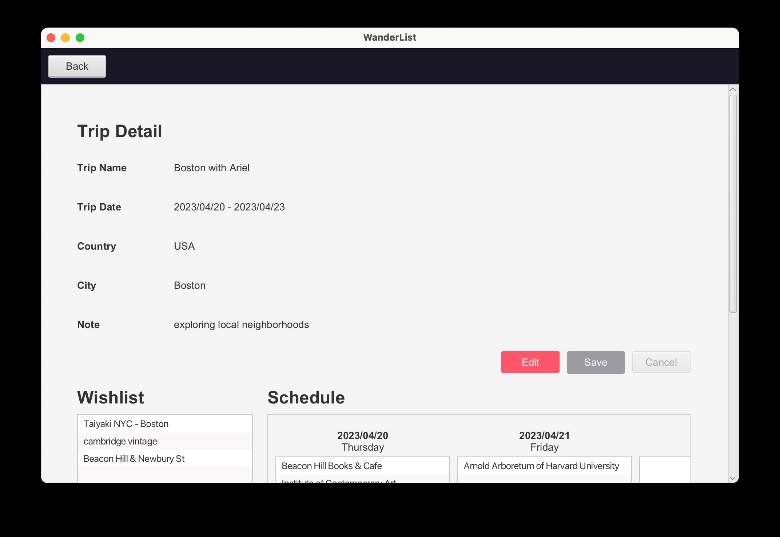


Figure 8. View trip details

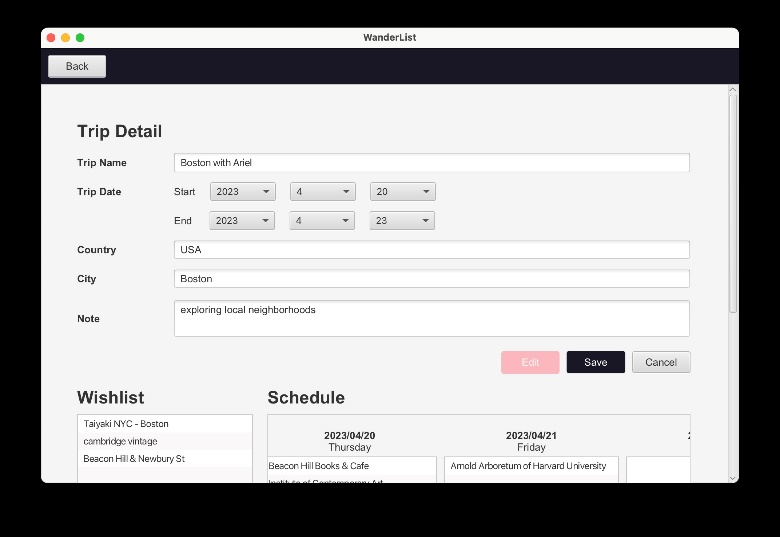


Figure 9. Edit trip details

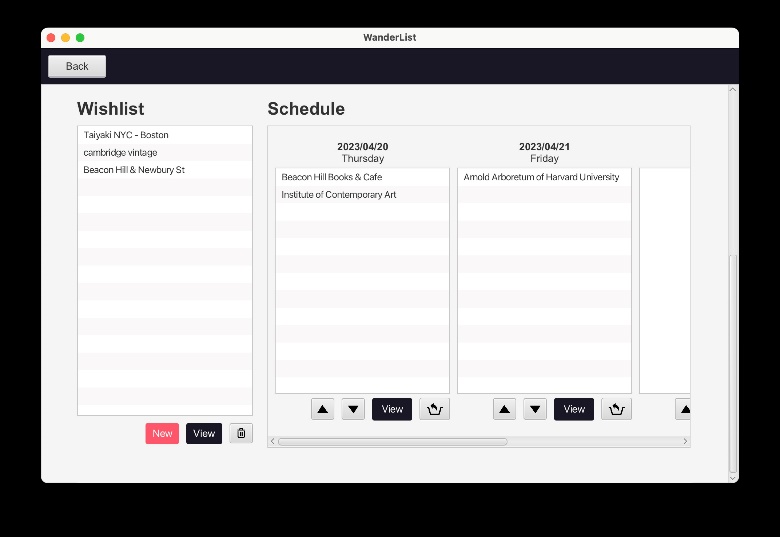


Figure 10. Wishlist and day view. Day view generated according to the trip dates. Users can drag and drop items to the day they wish to schedule the item. If the place does not operate on the assigned day, an alert will pop up.

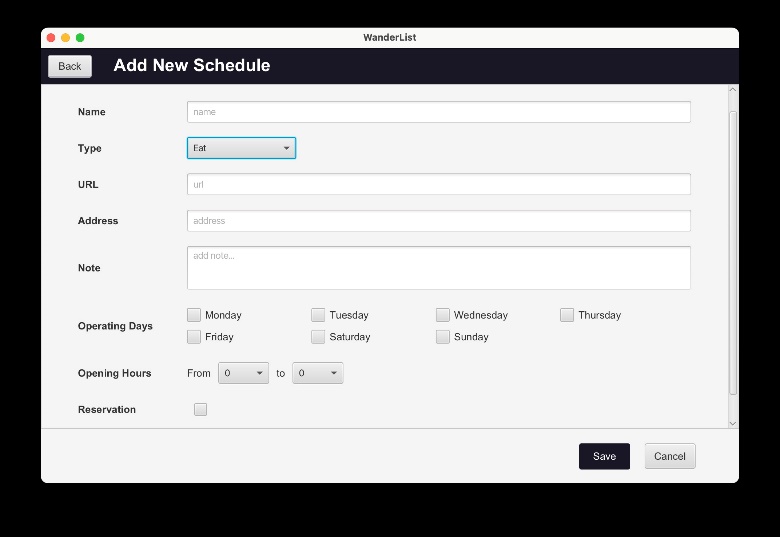


Figure 11. Add new schedule item. The page will load its specific properties when the user selects a type.

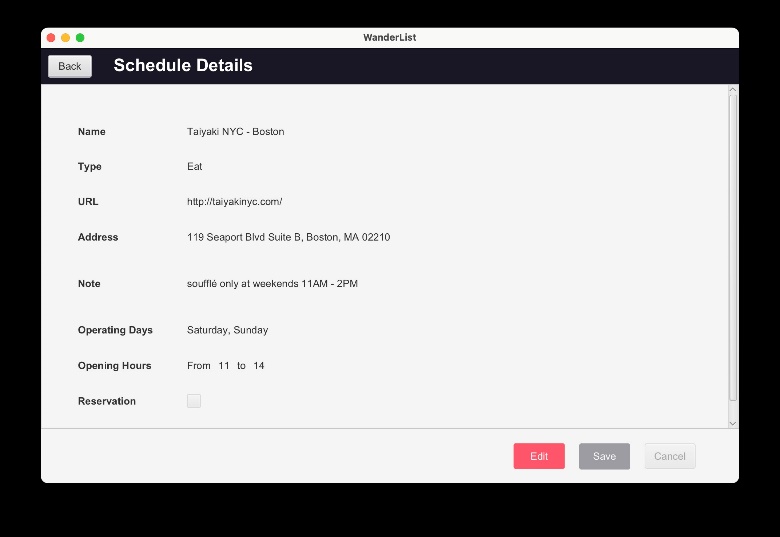


Figure 12. View item details

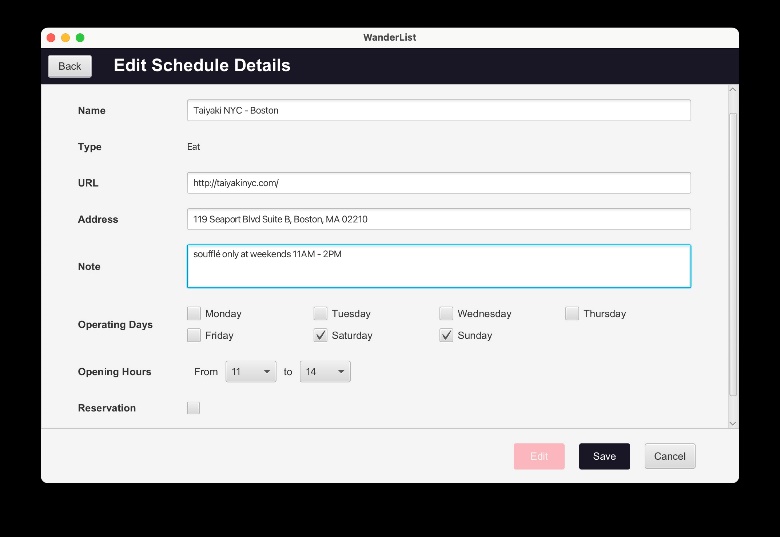


Figure 13. Edit item details

# **VII. Discussion (Reflection)**

After finishing the project, we found that there is a more efficient way to add, view, or edit schedule items by different types. In our project, we create different fxml files and controller java files by four different item type – eat, buy, play, and see. It causes the same work to be done 4 times. For example, if we try to adjust the layout or change the color of a button, we need to make sure all 4 fxml files changed. Or if we try to add an action on a button, we need to edit all controller files of 4 types. So, we discussed that we should create another controller file that allows 4 controllers of types to extend. After that, it will reduce the workload and save time on doing the same work.

# **VIII. Conclusions and Future Work**

ADVANTAGES

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 reservations. Also, users will not encounter a situation where the restaurant or shop is not open when they arrive.

PROBELM NOT EXPLORED

Some of the topics not explored in the projects are the inclusion of accommodation details, flight details, and packing list. In the future, we will add the functionality of integrating accommodation and flight information to add more types of projects based on the current ones. Additionally, the app should allow the user to list packing lists and ensure that the user has everything they need to travel.

IMPROVEMENT

We plan to enhance our project by adding two features. First, a travel-sharing feature for easy itinerary sharing with friends and family, promoting coordination and safety. Second, integrating Google Maps for real-time location data, directions, and nearby attractions exploration. These improvements will create a more seamless and enjoyable experience, making WanderList an essential travel planning tool.

# **IX. Job Assignment**

* Sheng-Jung Chen: Database connection, Dashboard page, Add and edit trip, Wishlist item scheduling
* Yu-Fang Chang: Add and edit items page, Wishlist item scheduling, UI beautify
* Yifei Chen: Login / Signup page and login authentication, Item types detail display pages, Presentation slides

##### **References**

1. Jung, Y., & Lee, Y. (2018). A Study on the Development of a Mobile Application for Personalized Travel Planning. Journal of Digital Convergence, 16(10), 157-167.
2. Ghazali, R., & Yusof, Y. (2020). Developing a Mobile Application for Travel Planning Using Java. Journal of Telecommunication, Electronic and Computer Engineering, 12(2-6), 149-153.
3. Scene Builder. (n.d.). Retrieved from <https://gluonhq.com/products/scene-builder/>
4. Oracle. (2022). JavaFX. Retrieved March 20, 2023, from <https://openjfx.io/>
5. Deitel, P. J., & Deitel, H. M. (2017). Java: how to program. Pearson.