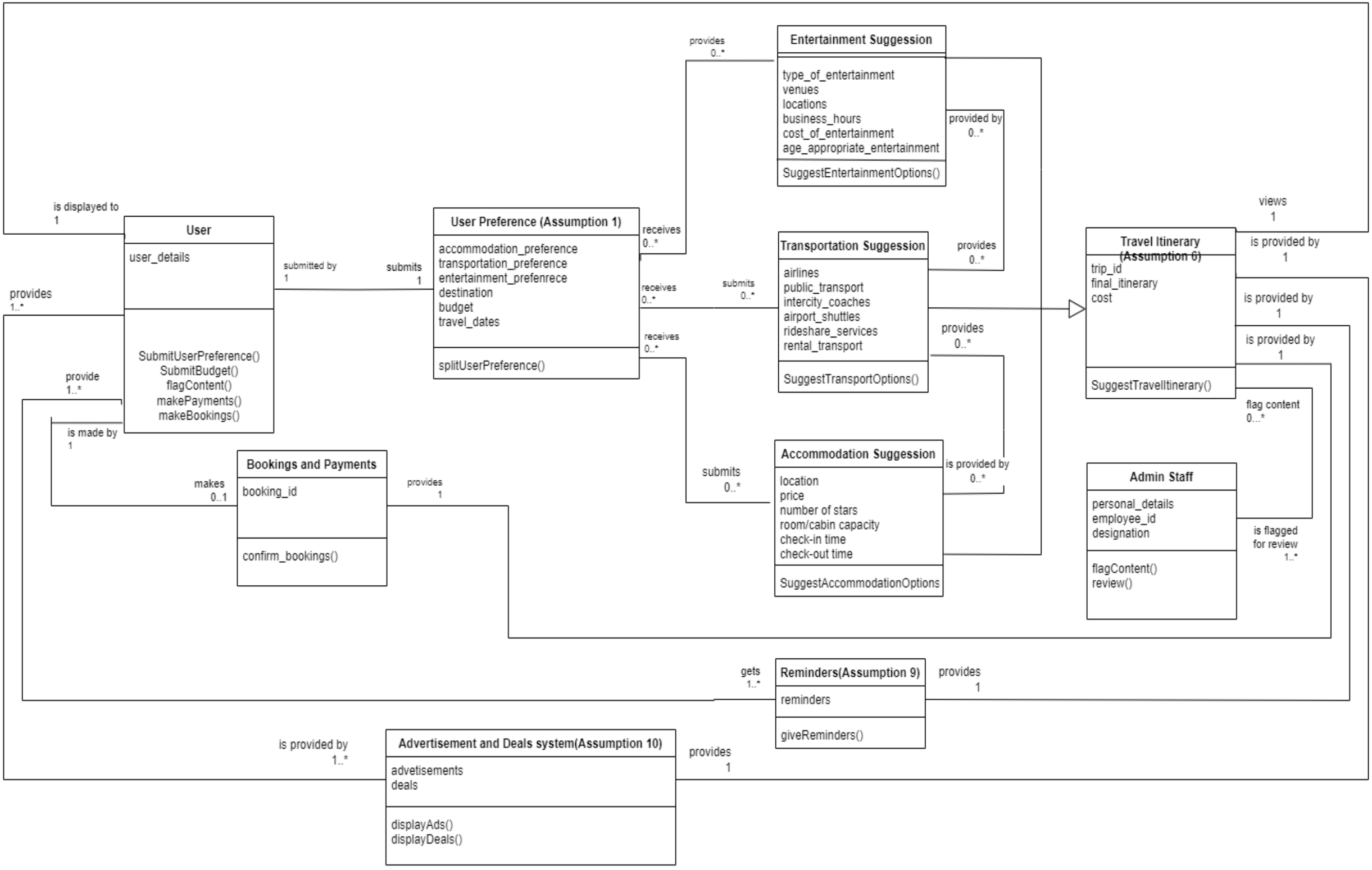
1. Additional Research and Assumptions

For this particular case study, we have taken an inspiration from the working of a website called "MakeMyTrip" It's an Indian online travel company founded in 2000. The company provides online travel services including air tickets, domestic and international holiday packages, hotel reservations, railway and bus tickets". We have referred to two amazing case studies on "MakeMyTrip" (Wane 2013) and (Barapatre 2014) available on the internet and have built our assignment accordingly.

Assumptions:

- Assumption 1: user preference covers everything from dates and times of travel, overall budget, accommodation preference, travel preference, entertainment
- Assumption 2: The integrated travel system will split the user preference into 3 categories: user accommodation preference, user transport preference and user entertainment preference and sent these three preferences to accommodation system, transport system and entertainment system respectively.
- Assumption 3: user accommodation preference covers everything from type of accommodation, budget, location, facilities, stars, room capacity
- Assumption 4: user transport preference covers everything from mode of transport, budget, facilities, preference and any other preferences as stated by the user
- Assumption 5: user entertainment preference covers everything from type of entertainment, budget, location, facilities, dates, time, age
- Assumption 6: Travel Itinerary will contain all the recommended travel, accommodation and entertainment options based on the user preference
- Assumption 7: There is a 'Manager of Travel Company' who manages the employees.
- Assumption 8: The travel company taken under consideration is a small to medium scale company. It has around 10 employees and 1 manager.
- Assumption 9: The system will send a reminder to user 8 hours before the time of travel or entertainment. e.g.: If user has a flight on Monday at 22:00 hrs, the system will send a reminder to user at 14:00 hrs.
- Assumption 10: There is a separate "Advertisement and Deals System" which will have information about all the ads and deals on offer. It will generate relevant ads and deals for user based on the user preference.
- Assumption 11: User cannot use the Integrated Travel System just to book stand-alone options of either transport, accommodation or entertainment options. User has to select a minimum of one transport, one entertainment and one accommodation option to proceed.
- Assumption 12: External systems means the external accommodation, entertainment and transportation system

2. Class Diagram

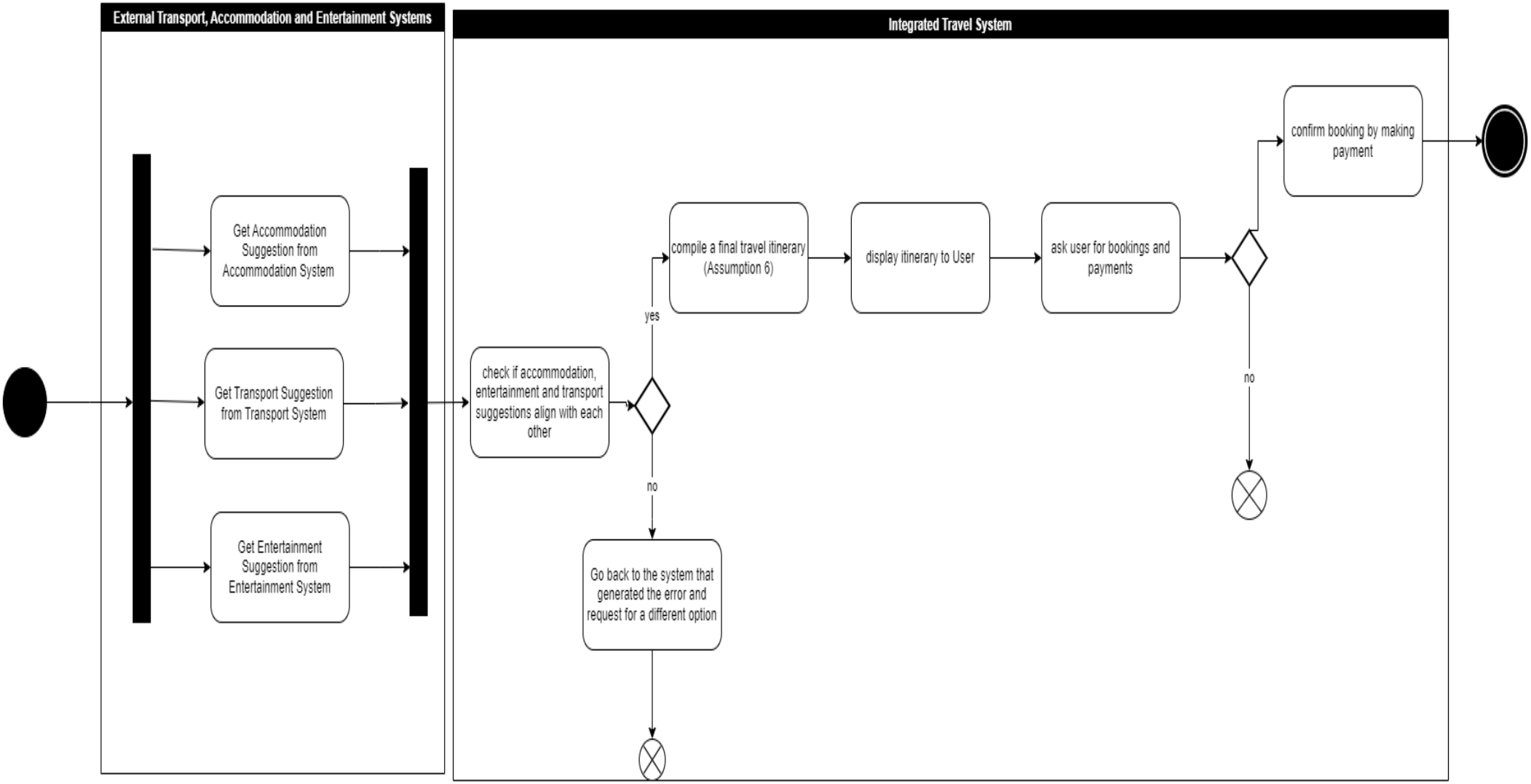


From the class diagram drawn above, the interaction between the ‘User Preference’, ‘Entertainment Suggestions’, ‘Transportation Suggestions’, ‘Accommodation Suggestions’ and ‘Travel Itinerary’ class is interesting. The ‘User Preference’ class invokes a method called ‘splitUserPreference() and provides accommodation_preference, transportation_preference, entertainment_preference, destination, budget, travel dates to the respective ‘Entertainment Suggestions’, ‘Transportation Suggestions’ and ‘Accommodation Suggestions’ classes. The case study makes it very clear that the transport needs to be in alignment with the check-in check-out times as well as the location and operation timings of the Entertainment Suggested. To reflect this requirement in the class diagram, we have included a association between the ‘Entertainment Suggestions’ and ‘Transportation Suggestions’ classes. We have also included an association between the and ‘Accommodation Suggestions’ and ‘Transportation Suggestions’ classes.

The multiplicity for the association is ‘0..*’. This means that the Accommodation Suggestions class can provide multiple options of accommodation and for each option the transportation suggestion class has to make sure that adequate transportation arrangements have been made. The multiplicity also tells us that the user can go forward with not using the system for accommodation booking. For example, if a user plans a trip where he is going to stay at a friend’s place, he can use the system to do the other bookings like flight, entertainment, etc. That’s when no output is going to come from the accommodation suggestions class as the user has not used the functionalities of the same.

Finally, the Entertainment Suggestions, Transportation Suggestions and Accommodation Suggestions classes are generalized into one class called ‘Travel Itinerary’. This class with contain all the information from the 3 classes. The integrated travel itinerary will be then displayed to the user.

3. Activity Diagram

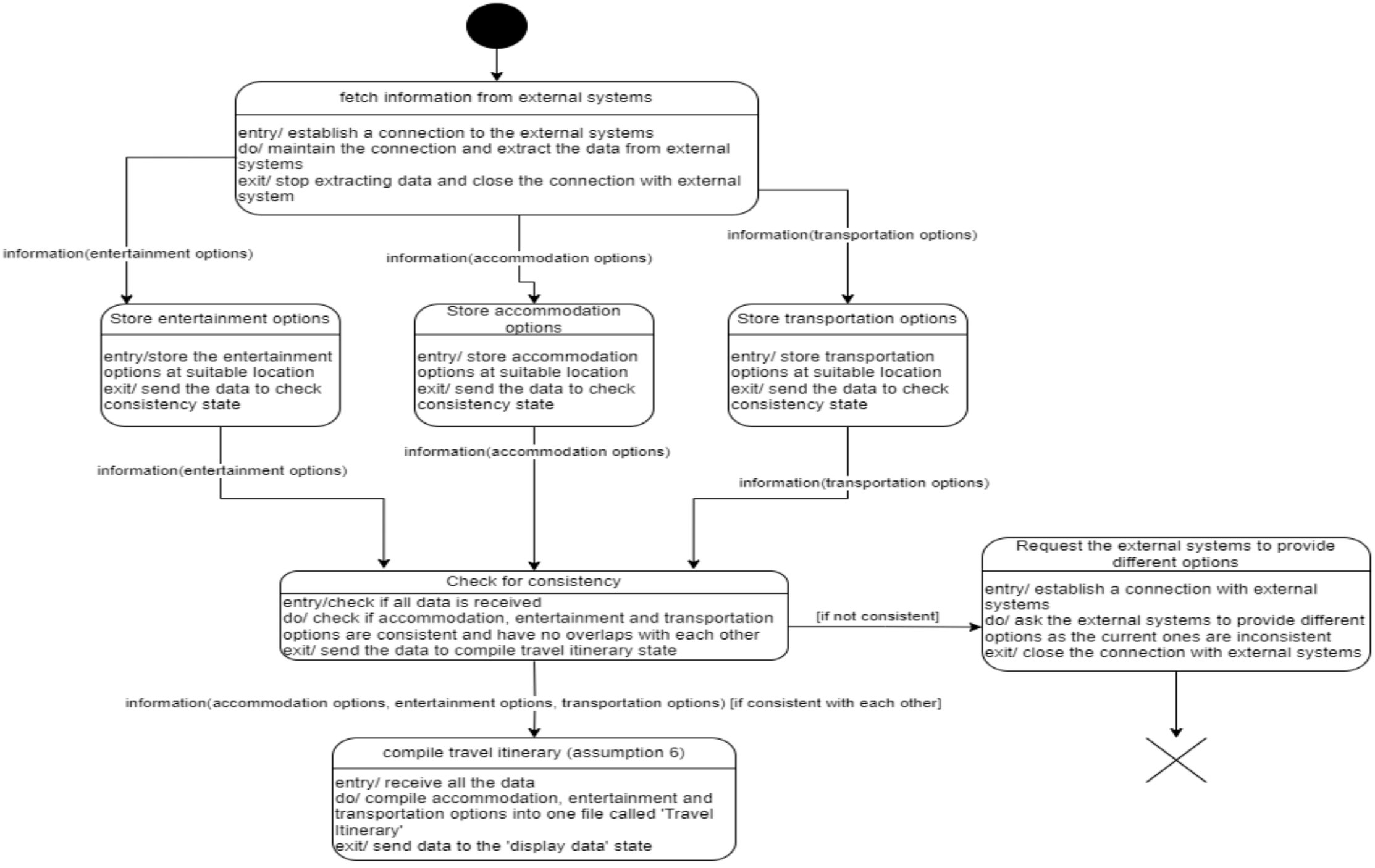


The activity chosen the draw the activity diagram focuses on the part of the system that integrates the information received from the external transportation, entertainment and accommodation system and then displays a compiled ‘final travel itinerary’ to the user. The activity diagram begins with the actions ‘Get Accommodation Suggestions from accommodation system’, ‘Get Transportation Suggestion from Transportation System’ and ‘Get Entertainment Suggestion from Entertainment System’. We have used synchronization because all the three activities will happen at the same time synchronously.

The output of this synchronization will go into another activity called ‘check if accommodation, entertainment, transportation suggestions align with each other. Then we have added the decision point with 2 flows coming out of it. If the output of activity is false i.e the 3 suggestions do not align with each other, the system will go in another action called “go back to the system that generated the error and request for a different option”. Then it will flow into a flow final node and that particular flow will end. However, the other flow from the decision point will continue and will flow into an action called ‘compile a final travel itinerary’. This will then flow into another action called ‘display itinerary to user’ and then it will flow into another action called ‘ask user for booking and payments.’

The user can use the system just to view the recommendations generated from the system and then leave. It is not mandatory for the user to go forward with the bookings. To accommodate that feature we have included a decision point after the ‘ask user for booking and payments’ action. If the user does not want to go forward, the system will go into a ‘flow final node’ and that particular flow will end. But if the user wants to continue with the bookings, then the system will go into another action called ‘confirm booking by making payment’ and will ultimately flow out of it into a ‘activity final node’. The activity final node indicates that the activity has ended.

4. State Machine Diagram

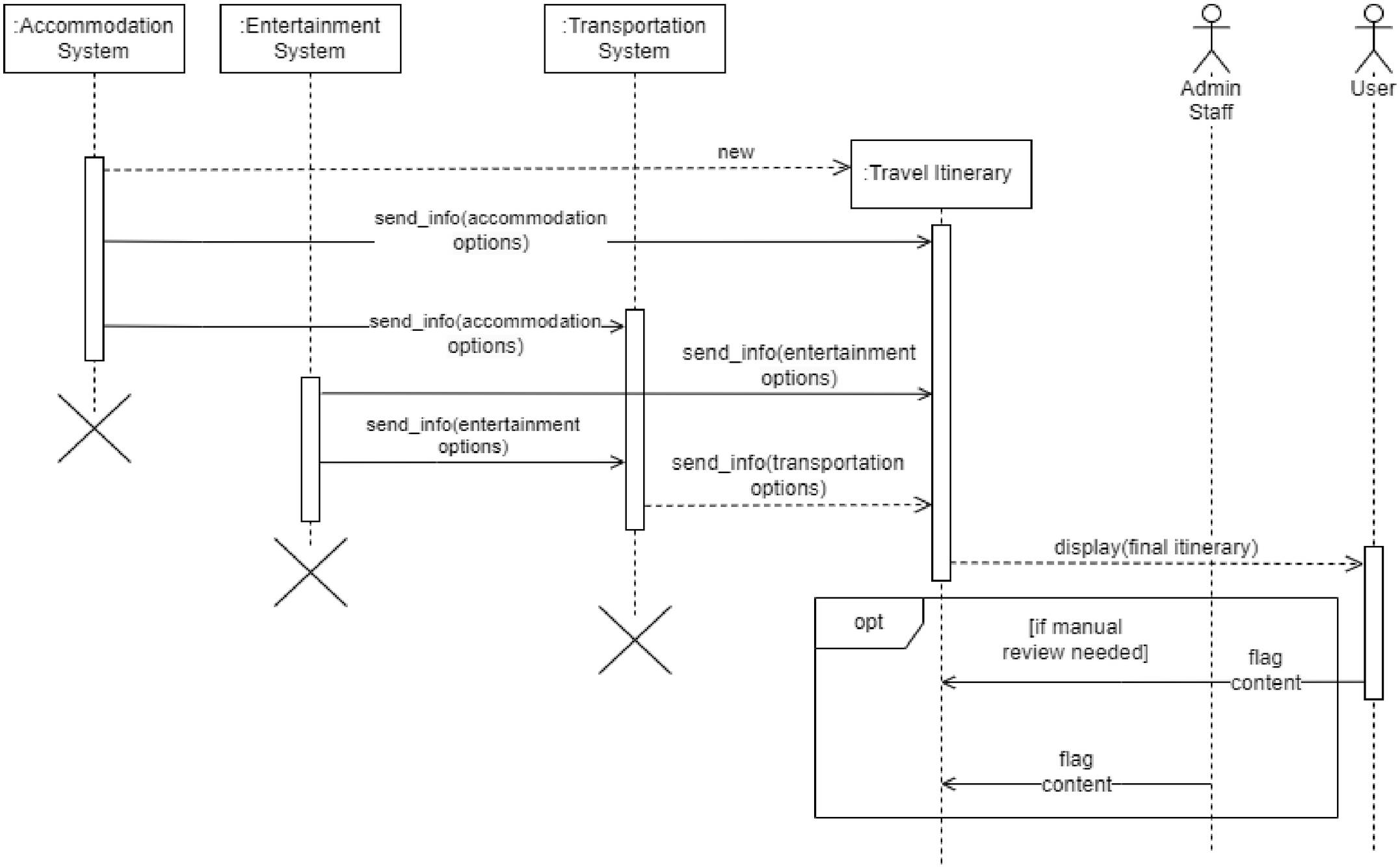


The class chosen to draw the state machine diagram is ‘Travel Itinerary’. When the process starts, the system goes into a state called ‘fetch information from external systems.’ In this state, the system will fetch the information from the other external systems like ‘Transportation’, ‘Entertainment’ and ‘Accommodation’ system. Then, it will send information about transportation options to ‘Store Transportation options’ state, ‘accommodation options’ to ‘store accommodation options’ state and entertainment options to ‘store entertainment options’ state.

In the ‘store accommodation options’ state, the system will store the information regarding accommodation options it has received into a database. Similarly, in the ‘store entertainment options’ state, the system will store the information regarding entertainment options it has received into a database. And in the ‘store transportation options’ state, the system will store the information regarding transportation options it has received into a database. Then the system will go into the ‘check for consistency’ state. In this state, the system will make sure that all the data it has received is consistent with each other. It should make sure that all the transportation options must be in sync with the check-in, check-out times. Also, there should be a provision of transportation to and from the entertainment location.

If everything is consistent (shown by the guard condition on transition), the system will go into a state called ‘compile travel itinerary’. Here the accommodation, entertainment and transportation options will be compiled by the system into one single file and then this file would be then displayed to the user. Then the system will go into a final pseudostate. If the system finds some inconsistencies with the accommodation, entertainment and transportation options, then it will go into a state called ‘request external systems to provide different options’ state. Here the system will establish a contact with the external transportation, accommodation and entertainment systems and will request them to provide new and different options as the current options are inconsistent. Then it will go into a terminate pseudostate.

5. Sequence Diagram



Initially the ‘Accommodation System’, ‘Entertainment System’ and ‘Transportation System’ objects are created. The accommodation system sends a message that creates a new object called ‘Travel Itinerary’ is created. Then it sends a message called send_info(accommodation_options) to the Travel Itinerary. The case study makes it very clear that the transport needs to be in alignment with the check-in check-out times as well as the location and operation timings of the Entertainment Suggested. We have reflected this in our sequence diagram. Accommodation system will send its info about accommodation options to transportation system. Then, the accommodation system object will destroy itself.

Entertainment System will send entertainment options to travel itinerary. Then it will send the same to transportation system. All the messages so far are asynchronous. However, the message titles send_info(transportation options) is a synchronous message as the system will wait for the messages coming from entertainment and accommodation system to transportation system to deliver before sending the send_info(transportation options) to travel itinerary. This provision is made because the transportation options must be aligned with the accommodation and entertainment options.

After all the data is received from transportation, entertainment and accommodation systems, the travel itinerary will send the data to the user object to view. This message is also synchronous. As mentioned in the case study, the user and admin staff has a feature to flag content if manual review is needed. This optional functionality is included in the opt fragment with a guard condition that basically tells that the fragment will be executed if manual review is required.

6. References

Dr Luke Nguyen-Hoan (2023) 'Class Diagrams' [lecture PowerPoint slides], *Systems Analysis and Modelling G 6677*, accessed 29 April 2023.

Dr Luke Nguyen-Hoan (2023) 'Activity Diagrams' [lecture PowerPoint slides], *Systems Analysis and Modelling G 6677*, accessed 29 April 2023.

Dr Luke Nguyen-Hoan (2023) 'State Machine Diagrams' [lecture PowerPoint slides], *Systems Analysis and Modelling G 6677*, accessed 29 April 2023.

Dr Luke Nguyen-Hoan (2023) 'Sequence Diagrams' [lecture PowerPoint slides], *Systems Analysis and Modelling G 6677*, accessed 29 April 2023.

Seidl, M. (2015) *UML @ classroom: An introduction to object-oriented modelling*. 1st ed. Cham, CH, Springer. doi:10.1007/978-3-319-12742-2.

Satzinger, JW, Jackson, RB & Burd, SD (2016), *Systems analysis and design in a changing world* Seventh edition., Cengage Learning, Boston, MA.

Rahul Wane (2013), *slideshare.net* [website], https://www.slideshare.net/rahulwane/makemytripcom-case-study-and-analysis, accessed 07 April 2023.

Ashish Barapatre (2014), *slideshare.net* [website], https://www.slideshare.net/ashishmindfreak/make-my-trip-31155894 accessed 07 April 2023.