Sparrow – A Travel Itinerary Recommender System for Group Preferences

Victor Chien
Jesse Cui
Alexander Lichen
Akshay Malhotra
Linzhi Qi

1 Abstract

Planning group travel can be a time-intensive and frustrating process. Aside from the high information and tracking costs associated with having to use different resources to book travel, lodging, dining, and attractions, the main issue with group trips is preferential conflicts. Individual members have different tastes in terms of everything ranging from destinations and dining to budget and schedule intensity. Reconciling all of these differences often costs significant energy and leads to unsatisfying compromises for everyone. As a result, a relaxing trip can often times, ironically, be prefixed by stress and frustration.

In order to lower the barriers to travel for millions of people around the world, we present Sparrow: an innovative and intuitive mobile application that consolidates member preferences to generate a trip itinerary cognizant of individual constraints and optimized for overall fairness and satisfaction. We have developed an application that relies on a unique fairness algorithm that helps resolve preferential conflict in groups with a goal to maximize overall satisfaction. In our evaluation, we use economic utility models and simulations to show the improvement our algorithm offers over standard methods like naïve voting. This work is also generalizable to the greater group decision-making space.

2 Motivation & Product High-Level Functionality

2.1 The Problem

The problem we're addressing is the following: **Planning travel can be a time intensive and frustrating process**. This problem can be broken down into two main components.

1. **Social/Preferential Conflicts**: Currently there are no intuitive ways for groups of friends to plan trips together and handle all the associated logistics before and during the trip. These problems were derived from personal travel experiences and the travel experiences of our friends. Members of a trip often have to spend a significant amount of time meeting in person to identify and resolve all of their differences.

2. **High Information Search and Tracking Cost**: People have to visit multiple websites to book lodging, make restaurant reservations, buy tickets to local attractions, book flights/buses etc. Keeping track of all of these different pieces of information can be a pain.

Aside from this, the problem also contains a large social impact component. Many publicly funded destinations, such as cultural sites and national parks, suffer from low exposure and subsequently monetary support for upkeeping purposes. Through Sparrow, we want to make users more aware of these destinations and allow them to easily contribute to their upkeep when they visit.

However, this problem can also be generalized to group decision-making. The main problem we address is social/preferential conflicts in group travel, but the setting can be generalized from group travel to include all of group decision-making. Any setting with groups and preferential conflict offers a potential application of Sparrow.

2.2 The Solution

Our solution, is an intuitive mobile application that generates travel itineraries for groups of varying sizes. To clarify, a travel itinerary in our case consists of a timeline spanning one or multiple days. Each day is filled with travel destinations, restaurants, etc. (ex. amusement parks or scenic beaches). The application will take into consideration the preferences by having users select from a series of options.

After all preferences are submitted, Sparrow will make API calls to the Google Maps API to retrieve a list of viable locations. Then Sparrow will further curate the destinations to maximize satisfaction of each member's preferences through our unique Fairness Algorithm. At the end, we present an itinerary for the group trip that offers a high degree of satisfaction for the most members possible.

2.3 Alternative Applications

In this project, we focus on a specific application of our solution, particularly group travel. However, any setting with groups and preferential conflict offers a potential application of Sparrow. For example, one area that we have also considered is budget planning for businesses, especially non-profit organizations. Often, there are a lot more items that management teams want to include in the annual budget than the actual capacity, so a slightly altered form of our algorithm could be applied in this setting. It can be used even for project planning because there are a lot of disagreements in the early stages when defining the scope. Consequently, a major alternative market is productivity for businesses.

Aside from a corporate setting, a similar algorithm could also be used for personal purposes. For example, we talked with one of our advisors about how there are family disagreements about where to eat out whenever they go, so we could apply our solution to achieving maximal satisfaction across a time series. The main point is that there are countless potential markets for this product. While we think the travel market is one that could really benefit from Sparrow, the main innovation lies in our algorithm, so there are many other applications.

3 Related Work

When surveying the travel landscape, we found two major problems in existing solutions. First, little thought is given to planning group trips in particular, despite 65% of people saying that the reason they travel is "to strengthen bonds with friends and family" (ustravel). Currently, the predominant solution is to enable itinerary sharing – this is the path that popular trip planning app TripIt currently takes. However, we believe such a solution is inadequate, as it does not solve the fundamental problem of coordinating different preferences amongst travelers in a group. A quick survey of the top Google search results for "plan a group trip" shows that all results mention the social difficulty of coordinating a group trip, including the aptly titled New York Times article, "Traveling with a group? Here's how to plan and stay friends". Conversations with peers and our own past experiences have shown that the coordination required to plan a group trip is immense and can often lead to frustration and arguments. Such a process would benefit from a more comprehensive solution.

Second, planning a trip currently involves a constellation of fragmented services. The first category of services includes companies such as TripAdvisor, Expedia, Lonely Planet, and Google. These sources function primarily as information providers - that is, they supply relevant material about where to go, what sites to see, and where to eat and stay. The second category of services are bookkeeping applications for managing trip itineraries - TripIt, TripCase, and basic spreadsheets fall into this category of applications. There's then a 3rd category of services that handle a variety of miscellaneous tasks - currency conversion, maps, etc. We find that coordinating between all these services is a challenge, especially when balancing preferences of different individuals on a trip.

Other related work is more technical in nature. Over the course of our research, a number of academic papers have proved useful. Roopesh and Talushi (2019) provide a survey of travel recommendation algorithms. Ravi and Vairavasundaram (2016) provide a more-detailed look at machine-learning based approaches. For fairly balancing user preferences, a paper by Serbos et. al. (2017) entitled Fairness in Package-to-Group Recommendations provides a useful example.

Other external work we've relied upon includes 3rd party APIs. We decided to use the Google Places API as this gave us the most flexibility from a Terms and Services perspective. The API provides us with critical content which we serve to users of the application.

4 Technical Approach

In this section, we present the technical details. The repository for our code can be found at the following links: github.com/vicoociv/senior-design-client (Frontend) & github.com/vicoociv/senior-design-backend (Backend).

4.1 Application

4.1.1 Frontend

Sparrow is a mobile application built using React Native, which allows it to run on both iOS and Android phones. Limiting the application to only one set of users would be limiting in terms of growing our user base.

4.2 Backend

Our backend is built using Firebase, a Google cloud and storage platform. There are two parts to our backend: cloud functions (endpoints) and database. The cloud functions run using a serverless framework, which means our backend services are highly scalable. The number of servers that host our backend will rise and fall with the amount of traffic we receive to ensure that we have the necessary bandwidth to give all of our users a smooth experience. We also only have to pay for the servers currently in use. For our database, we use Firebase FIrestore. This is Firebase's storage service and is highly scalable as well.

4.3 Middleware

To allow information to flow between our React Native application and our Firebase backend, we use the redux-saga library developed by Facebook. This library allows us to easily manage, efficiently execute, and transparently test asynchronous activities our app undertakes, such as data fetching and cache accessing. It also allows us to better handle failures, which in turn makes our application more robust overall.

4.4 Third Party APIs

We use several third party APIs, including Google Places, Google Places Autocomplete, and Google Places Photo. We use the Google Places API to retrieve tourist destination data. We use the Google Places Autocomplete API to make it easy for our users to denote which city they plan to visit when creating a new trip. This API also gives us the latitude and longitude of the city, which we need to feed into the Google Places API in order to retrieve the destinations in that city. Finally, we use the Google Places Photo API to get images for each destination. The Google Places API does not return associated images with each destination unfortunately, which means we have to use this additional API.

4.5 Design

To design our application, we used Figma, a collaborative, we-based design tool for user experience and user interface design. We were also able to create an application prototype on Figma to simulate real world intractability. Through extensive research and design-iterations, we modeled our design language and flow after that of several highly intuitive and popular mobile applications in the market, most notably Airbnb. The reasoning for this is that users will more easily understand how to use our application if we modeled certain user flows and design elements after what they already know how to and love to use.

4.6 Fairness Algorithm

One of the primary distinguishing features of our application is the fairness algorithm that attempts to maximize overall satisfaction for the members of a group trip. This problem is known to be quite challenging because different members of a trip often have very different tastes and preferences, thus making it very difficult to satisfy everyone.

Some interesting questions also need to be addressed from a philosophical perspective when designing such an algorithm. For example, for a group in which a small minority absolutely disagrees with the majority opinion, is it more fair to be inclusive of the minority opinion or to routinely follow the traditional democratic procedure? We attempt to address this issue and other related ones with a solution that takes into account multiple perspectives.

Achieving the optimal solution is a well-known NP-Hard problem, but we base our solution off an an algorithm that has been mathematically proven to be within a factor of $1 - \frac{1}{e}$ of the optimal solution. We take a greedy algorithm approach to maximize expected utility for each sub-problem and then aggregate the solutions to present a global solution.

We structure our itinerary construction process in the following manner: First, we choose destinations. Then, assuming we have those destinations, we select restaurants. Finally, assuming we have the destinations and restaurants locked, we determine lodging. Each previous stage is fixed before moving on to the next stage.

For the purposes of this analysis, however, we explore the destinations problem with the understanding that the other problems, such as restaurants, can be solved in a similar manner. More specifically, with each member having a set of certain destinations in his or her preference set, can we construct a fair set for the group to visit?

4.6.1 The Formalized Problem

Input

- Set S of all group members' destination preferences, with each S_j a set of member j's top n unordered selections
- m representing number of destinations that final itinerary should contain with $m < n$
- λ representing group's perceived importance of preference selection relative to traveled distance

Output

- Set F consisting of m destinations for the group to visit such that the collection of destinations is fair

4.6.2 The Algorithm

1. Create and initialize map W of each destination i to weight w_i, initially defined as number of votes
2. Initializing voting power v_j of each member j as 1
3. Remove destination i with max weight w_i from W and add to F
4. Set voting power of each member j with destination i in his or her preference set as $v_j = v_j \cdot \frac{1}{k+1}$, where k represents total number of members with i in their preference sets
5. Compute preference weight of each destination i as $p_i = \sum_{j \in J} v_j \cdot P_{ij}$ where P_{ij} is an indicator variable representing whether $d_i \in S_j$
6. Standardize p_i across all destinations as pz_i by dividing with the maximum value
7. For each destination i in W, compute minimum Cartesian distance d_i between it and some element of S using latitude and longitude
8. Standardize d_i across all destinations as dz_i by dividing with the maximum value, subtracting 1 and computing the absolute value so as to effectively compute a standardized inverse
9. Update weight w_i for each destination i in W as $w_i = \lambda \cdot pz_i + (1 - \lambda) \cdot dz_i$
10. Repeat steps 3-9 until the size of F is equal to m

4.6.3 Basic Properties

Before we analyze the performance of the algorithm, we begin by pointing out some basic properties. First, we should note that we define fairness on a philosophical level from the utilitarian perspective. On a mathematical level, this means we are aiming to maximize aggregate expected utility across all group members.

With respect to this goal, there are a few interesting features we see. First, we introduce a λ parameter to this algorithm in order to balance the tradeoff between preference maximization and traveled distance. To motivate this, let us consider an extreme example. Assume we are planning a trip to the United States, and the destination preferences in order of most votes are the Statue of Liberty, the Grand Canyon and the Empire State Building. If we are to create an itinerary with 2 of these, most people would choose the first and third because they are much easier to reach. Particularly, the positive differential in expected utility by visiting the Grand Canyon is far outweighed by the negative differential in expected utility due to the travel time and costs.

Another feature is the updating mechanism for voting power. Every time a group member's preference is selected, his or her voting power decreases. However, it is important to note that the voting power change is inversely proportional to the number of people who selected the destination. This means that selecting a minority preference costs a lot more voting power than it does to select a majority preference destination. In addition, we should note that under this updating mechanism, the voting power asymptotically approaches 0 without ever actually touching it, thus ensuring no agent is completely excluded from the voting process at any point.

Finally, we present the following proposition: Under the absence of asymmetric information, no player can predictably game the system by misrepresenting his or her preferences. Let us prove this. Assume toward a contradiction that some player can achieve a higher expected utility by misrepresenting his preferences. Then he must not select at least one of his actual preferences X and select at least one of his non-actual preferences Y. When he selects Y, he must know that Y will not fall in the output because otherwise it would lower his expected utility. However, this would imply that he knows at least some other player's preferences, thus violating the condition of the absence of asymmetric information. Alternatively, it must be that he knows X will fall in the output set regardless, but then this also implies the existence of asymmetric information, thus providing a contradiction and completing the proof.

It is critical to make the distinction that, in actuality, there is likely to be some asymmetric information depending on who group members discuss their preferences with before the execution of the algorithm. However, there is no practical algorithm that can account for this. We must make the assumption that each player is aware of only his or her own preferences, and our analysis crucially relies on this concept.

5 Evaluation

The initial plan for evaluation was to test it with groups traveling over Spring Break. They would use the application and rate it on a variety of features, but due to the COVID-19 pandemic, we had to alter our evaluation process.

5.1 Fairness Algorithm

It is often difficult to ascertain utility in experiments with live users due to the challenge of assigning a numerical value to happiness, subconscious biases, and other related factors. As a result, we begin our evaluation process with a simulation of our theoretical model of the representative agent's utility.

5.1.1 Utility Representation

Assuming the satisfaction of von Neumann-Morgenstern utility conditions, we define our multivariate utility function as follows:

$$U(p,d) = \lambda \frac{p^{1-\eta}-1}{1-\eta} - (1-\lambda)\frac{1-e^{-\gamma d}}{\gamma}$$

So, for a given number of an agent's preferences that are present in the output set p and the total distance traveled to cover all the locations in the output set d, we compute each agent's expected utility with the parameters λ, η and γ.

We model utility similar to a standard consumption process, particularly as a sum of two independent processes. The first part of the equation is derived from a standard iso-elastic (power) utility model with the parameter η representing preference risk aversion, or more specifically, the degree to which the agent would prefer the status quo number of preferred locations in the itinerary than risking the loss of one in order to gain an additional one.

The second part of the equation is derived from a standard exponential utility model with the parameter η representing distance risk aversion, or more specifically, the degree to which the agent would prefer the status quo traveling distance in the itinerary than risking the addition of one more unit of distance in order to lower it by one unit. Of course, each component is weighted by the λ parameter, which represents the relative importance of each for the agent.

We now turn to the first-order and second-order derivations to discover some additional interesting properties about risk aversion in this utility function. The complete derivations can be found in Appendix A, but the important point to note is that preference has decreasing absolute risk aversion, while distance has constant absolute risk aversion, and this is by design.

For preferences, it makes sense for it to display decreasing absolute risk aversion because after a certain point, once the agent has achieved a high base level of utility, he or she is less risk averse since a high level of utility is already present and due to declining marginal returns.

On the other hand, it is unclear what the absolute risk aversion for distance should be. Some might suggest that after a certain level of travel distance has been reached, they are less risk averse because they already know they will have to travel far more than their taste. So, they would value less travel more than they would suffer from more. However, others might argue for increasing absolute risk aversion since after that point has been reached, they should be even more risk averse, particularly given that any further exertion might cross the breaking point for them. Since it is unclear, we average this out and select constant absolute risk aversion.

5.1.2 Simulation Setup

We begin by setting the parameters. Based on empirical analysis, the chosen values for η and γ are $\frac{1}{2}$ and $-\frac{1}{2}$, respectively. In addition, we have empirically tested and selected a value of 0.75 for λ. Values of λ close to this also perform similarly.

For each agent, it is easy to assess the value of p based on how many of their preferences are in the final output set F. On the other hand, the distance problem is more challenging since the set is unordered. So, we must find the minimum travel distance necessary to visit every destination. We can clearly see that this problem is basically the Traveling Salesman Problem, which is obviously exponential time, but since our input size is very small in every reasonable case, we can simply use a current implementation of this problem to obtain our value of d.

The other parameters for the problem will be an option set of 10 destinations and an itinerary size of 5 destinations. We will vary the number of members in the group and the number of options each selects. We will run 1,000 simulations for each parameter value holding the other constant. The starting standard values will be 5 members in the group and 7 destinations to select.

Using these values, we can compute each group's aggregate utility as $\sum_{j \in J} U_j$ and compare it with standard approaches to solve this problem, such as the naive method of pure democratic voting for the top m preferences. Particularly, we measure percentage improvement in terms of computed aggregate utility.

5.1.3 Simulation Results & Analysis

In the base case, we observe an improvement of roughly 78% over naive voting with our algorithm. Below are the detailed results of the simulation analysis when we vary some of the parameters of the problem.

As shown in Figure 1, the performance of the algorithm improves even further compared to the naive algorithm when the size of the group increases. Of course, this follows by design since we want this algorithm to be fair in large settings.

As can be seen in Figure 2, the performance of the algorithm peaks at a specific ratio of preference set size to number of destination options. This again follows by design because if this ratio is too low, then it is unlikely that the algorithm will find much overlap between preferences, and if the ratio is too high, then even naive voting will perform well.

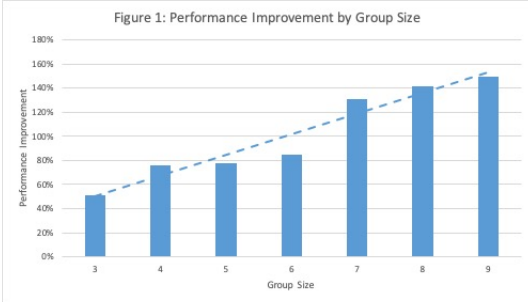

Figure 1: Performance Improvement by Group Size

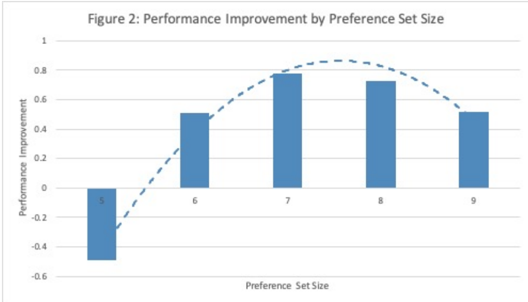

Figure 2: Performance Improvement by Preference Set Size

Overall, the simulation analysis and overall evaluation of this algorithm show that this improved fairness algorithm performs as expected and beats the current options.

5.2 Application

The other major component is the UI of the application, and the evaluation for this component depended primarily on understandability, ease of use, and aesthetic. In order to understand how our application fared on these metrics, we asked potential users to rate it on a basic 1-3 scale on each of them. The results showed that understandability and ease of use was high, the average score for aesthetic was lower initially, so there is still some scope for improvement there. We have refined it further since then.

6 Societal Impact

Technology has impacts. Sparrow seeks to be a responsible custodian of our users well-being. To do so, we take the privacy, security, and safety of our users extremely seriously.

For privacy, our goal is to take specific measures ensuring that our users have control over their own information. We will anonymize all data provided to third parties, provide users with audit logs about what data Sparrow collects and shares, and promise clear communication in line with GDPR best practices. For security, we implemented secure login through firebase authentication, which also helps us handle malicious attacks such as DDOS attacks. We also will follow general security best practices, including internal controls on who can access what data, encryption of important information, and more. Finally, for safety, we know that traveling internationally can be scary. In the event of unsafe conditions, Sparrow will work with its users to provide relevant information and help extricate them from unsafe situations.

One further consideration is anonymity. We ensure anonymity within the application when each user selects their preferences, so that no user worries about sharing private information. For instance, if an individual is self-conscious about their own budget limit, that information provided to our algorithms is automatically anonymized and the other travelers on the trip will not be able to see it, ensuring anyone who feels vulnerable about some aspect of their own preferences will be anonymous.

7 Business Plan

We present our business analysis in this section. For more details on the problem overview and competitive analysis, please refer to the motivation and related work sections, respectively.

7.1 Value Proposition

The project is more than a simple application for our potential users; we have a viable business plan to turn it into much more. For the casual traveler exploring new destinations with friends or family, Sparrow offers a uniquely painless experience in planning a trip suited for minimal bureaucratic coordination and conflict and maximal enjoyment.

For the management team of the destination attraction organically generate nearly as much traffic as it should, Sparrow offers an opportunity to be discovered by the masses and generate a level of interest in line with its potential. For the small, family-owned restaurant that presents a truly special experience and food cooked just like mom used to, Sparrow offers the chance to attract new customers.

7.2 Stakeholders

Aside from the stakeholders mentioned in the value proposition, the following are also important stakeholders:

- **Third-Party Sites**: We need to work with sites like Google Maps, Google Places, etc. to access destinations, hotels, flights, etc. to put in our itinerary recommendation. Most importantly, we need Google Maps' API to provide users with travel instructions/directions in between locations and Google Places to provide users with destination information. We will not have to interface with a majority of these third party sites for our MVP. We will do so in the long run. For the MVP, we will at most need to use Google's Places API to get data on travel destinations.

- **College Students**: Most college students take trips with friends after graduation or during school breaks. They often also have multiple preferences, such as affordability and varying diet requirements.

- **Tour Companies**: Tour companies can create custom trip packages for users in exchange for a small payment. This will help users save even more planning time.

- **Governments**: Governments of tourism-based countries can partner with Sparrow to encourage more people to visit their countries and boost their economies.

7.3 Market Research & Analysis

We conducted market research of 22 individuals within our target demographic. The results of our research shows not only a sizeable market of people who frequently go on trips, but also a significant need for a product that can aid with the trip planning process and can support group deviations. The appendix has detailed survey results, but notable quantitative research results include the following:

- Our target demographic often goes on trips. 36.4% of respondents have gone on a trip in the previous month. This number increases to 59.2% for the past three months and 77.4% within the past 6 months. 45.4% of respondents go on at least 3 trips a year.

- There is both a need for shorter and longer trips. 31.8% of users' most recent trip was over 1 week, and 36.0% of users prefer trips over a week in duration.

- Respondents go on trips ranging between 2 and 16 people, with a mode of 4.

- 36.4% of respondents were highly involved in their trip planning process.

- 36.4% of users sometimes or often deviate from their group.

Additionally, notable qualitative results from the survey include the following:

What do you dislike about the trip planning process?

- *Figuring out what everyone wanted to do— I want to do one thing but other person doesn't so overall bleh—- also there was only 5 spaces in car so we had trouble figuring out who to invite cuz people might feel bad for not being invited but also had to fill all five seats because car is expensive and must be split amongst most people possible (car was rented)*

- *Difficult to get people to meet at the same time to discuss logistics*

- *Too many options and not enough information about the best place to visit*

- *Time spent searching web for things to do and places to eat*

- *Annoying to coordinate with others*

If there was an app to help college students travel for fun, what should it do?

- *Coordinate availabilities and list travel preferences for each member*

- *Predesigned suggestions for itineraries listing places to visit and places to eat based off of preferences that the students would input*

- *Provide customizable travel itineraries accommodating for preferences such as economics etc.*

We also converted results of their annoyances to a word cloud. The top words for our respondents when asked about annoyances in terms of trip planning are "time" and "people", both of which our solution aims to tackle. The top words when asked about what they would like to see in the app are "list", "places", and "preferences", showing a need for an app that can list places and take preferences You can view these word clouds in the appendix..

7.4 Revenue Model

Below are the main components of the revenue model.

7.4.1 Targeted Advertising

For a given destination and set of user preferences, our algorithm can place sponsored restaurants and attractions higher in the list of options we present to the user. Aside from the social value of this feature in that we are presenting smaller businesses with an opportunity to attract new customers who might otherwise just go to the more well-known options due to their lack of familiarity with the area, we can generate a strong source of revenue for our business. We still need to work out the exact details of this feature, such as how exactly attractions and restaurants would be billed. Possible options include billing on a per-suggestion basis, in which we would bill for every time our application suggests the location, or a per-itinerary basis, in which we would bill for every time a user actually goes there. Additionally, it is not necessary that this targeted advertising is limited only to restaurants and attractions; it could include other local businesses as well. Regardless, this is a strong revenue source in this business.

7.4.2 Data

Once we are at a stage to make sure we have the necessary legal paperwork to go forward with this feature, we can easily monetize the user data we have. It goes without question that there is value in this data. For example, we will have interesting data on what type of attractions people prefer, what food they like, what their budget is like, etc. With all this information, it would be quite easy to create a demographic profile for each user. In addition, we could determine relationships between what types of restaurants and attractions people prefer and what kinds of people prefer what exactly. These customer insights could be offered to our partners, along with the opportunity for targeted advertising. Finally, we will have users rate each destination. With this data, we can analyze how the ordering of destinations can affect the destination's ratings. For instance, if there are three destinations (A, B, and C), we can have users visit A first and then B right after across hundreds of trips and compare the experiences with visiting C first and then B second. If users rate the A-B destination combination higher than the C-B destination combination, we can determine that A and B positively affect one another more so than C and B. We can not only use this data to improve our itinerary recommendation algorithms, but also sell it to the various destinations to help them better understand their customer preferences and attract new customers.

7.4.3 Travel Packages

We can partner with travel agencies and independent travel guides to offer pre-curated travel packages to our users. A pre-curated package would include a full-fledge itinerary with destinations, restaurants, and lodging included. These partners will only need to pay us a monthly fee to feature their travel packages and users will only need to pay a few dollars to use them. Users can either use the packages as is, or further customize them.

7.4.4 Transaction Fees

Another potential source of revenue is to charge transaction fees. Using third party APIs, we can offer 1-click booking to our users across all their selected hotels and transportation, saving the user significant time. In return, we can charge a fee. This can either be a fixed fee based on the number of items booked, or a percentage fee based on the value of the fares purchased. In either case, we expect this to be a recurring portion of our income.

In this section, we present the business analysis. For an overview of the problem and a competitive analysis, please refer to the motivation and related work sections, respectively.

7.5 Cost Model

The primary costs associated with this business are server hosting expenses, which are expected to be relatively small compared to the revenues. These expenses, however, are expected to be recurring and will grow proportionally with our user base. We foresee one of our main long-term cost drivers to be marketing. Even though Sparrow will have an easier time than other social networks at achieving a network effect, we still need to build up a strong initial user base. As our users will not be traveling all the time, we will need to periodically market Sparrow during peak travel seasons. The reason for this is that some of our users will demonstrate less engagement on the social side of our app, leading to lower levels of general engagement.

In order to retain these users when they decide to take their next trip, we need to undertake marketing campaigns to remind them of Sparrow. The hope is that over time, repeated exposure will make Sparrow the default, go-to option for many of these travelers. In terms of technical development and design costs, our team's combined technical and design skills are more than sufficient for bootstrapping an initial product. As we grow into a larger platform, however, we will need to hire more developers and expertise in other functional areas. Similar to marketing, maintaining a full team of employees will also be a significant long term cost driver.

8 Discussions & Lessons Learned

Overall, we are proud of the work that we placed into developing Sparrow. We learned a lot of important and valuable lessons along the way that will help us become better engineers. A few of the lessons learned are listed below.

8.1 Agile Development in Practice

It's important to focus on developing and testing core features of your application before moving onto developing other features. In the beginning, we were ambitious and had many ideas for features, such as social media sharing and itinerary updating. However, these needed to be cut to focus on ensuring that our core itinerary generation was fully functioning and effective at being fair. Also, having a simpler core app made it easier to explain and pitch to testers and audiences, and also made it easier to evaluate the core app since there were less components to evaluate. This lesson of ensuring core components work before moving on to other components is corroborated Agile development cycle theory in the industry.

8.2 Delegation, communication, and accountability when working in parallel

Since there were many moving parts, to ensure we didn't waste time developing the same feature among different people, we split up the tasks based on what technologies we were experienced with so we could parallelize work-streams. However, this workstyle involves more accountability and communication. Full accountability was necessary to make sure no one was the bottleneck. Full communication was necessary to ensure that all functionality of individual components were understood clearly before being connected to the larger system.

Some components would depend on the completion of other components from someone else. For example, to ensure that the front-end could generate trips upon the last person giving preferences, we had to make sure our back-end had the full functionality to upload preferences and to know when the last person uploaded preferences so that an itinerary object could be generated in the database. However, the back-end could only be fully completed if the itinerary algorithm is completed in implementation, that could be only implemented if we finished the core math behind the algorithm. We had different people in charge of each step, and so if one person slacked off, the rest of the components would not have been able to be completed. Effective accountability and communication allowed us to ensure no bottlenecks would occur and that development was smooth and effective.

8.3 Understanding your audience

As M&T students, we had to pitch to two separate audiences. We at first used similar slides to pitch to both business and engineering audiences and evaluators, but soon realize what business people find important are not what engineering evaluators find important. For example, we realized business audiences focused more on our potential monetization and applications of our algorithm, whereas engineering audiences focused more on the algorithm itself. So later, we made sure to cater and customize our presentations individually to suit the unique backgrounds of our respective audiences.

8.4 Evaluating components

Since every component is interconnected to another component in a complex system, a failure of just one component can fail the entire app, so it's important we have rigorous evaluation standards and tests for each component.

Evaluating components came in two parts. One is evaluation of a completed system. This type of evaluation was what we presented in classwork, such as evaluating the responses to the UI/UX of our app as well as the evaluation of the utility gain from our model.

The other type of evaluation is within the technical components themselves during the software development process. As mentioned before, we have several dependencies of components. The front-end components rely on the database, the database relies on the backend, and the backend relies on the itinerary algorithm and the APIs working. Any component failing in this system would have significant downstream negative effects, and since at the end of the dependency chain is the front-end of the app, we would not be able to have a working app if any of the components failed.

Therefore, it was important for us to unit test as many of the components as possible. For example, we tested database updates with test data JSON objects to ensure writes and reads to the database and our back-end logic was working well. It was also important for us to communicate to others if there were any errors or issues with one component, so that others would know to work around such issues.

Acknowledgements

We would like to thank a number of mentors and advisors for their guidance throughout the first stages of this project. Particularly, we would like to extend our gratitude to Professor Swapneel Sheth for serving as our advisor, Professors Susan Davidson and Ani Nenkova for providing useful advice during faculty consultations, Dr. Sangeeta Vohra for helping us in the initial stages of our project, and Professor Saikat Chaudhuri for advice on the business-related aspects of our project.

9 References

Roopesh L R and Tulasi Bomatpalli. 2019. A Survey of Travel Recommender System.

Logesh Ravi and Subramaniyaswamy Vairavasundaram. 2016. A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. Computational Intelligence and Neuroscience.

A Risk Aversion Derivation

$$U(p,d) = \lambda\frac{p^{1-\eta}-1}{1-\eta} - (1-\lambda)\frac{1-e^{-\gamma d}}{\gamma}$$

We first consider the preference risk aversion part of the model.

$$\frac{\partial U}{\partial p} = \frac{\lambda(1-\eta)p^{-\eta}}{1-\eta} = \lambda p^{-\eta}$$

$$\frac{\partial^2 U}{\partial p^2} = -\eta\lambda p^{-\eta-1}$$

We use this to determine absolute risk aversion.

$$\text{ARA} = -\frac{\frac{\partial^2 U}{\partial p^2}}{\frac{\partial U}{\partial p}} = \frac{-\eta\lambda p^{-\eta-1}}{\lambda p^{-\eta}} = -\eta p^{-1} = \frac{-\eta}{p}$$

$$\rightarrow \lim_{p\rightarrow\infty}\frac{-\eta}{p} = 0 \rightarrow \text{DARA}$$

Now, let us consider the distance risk aversion part of the model.

$$\frac{\partial U}{\partial d} = (\lambda-1)\frac{\gamma e^{-\gamma d}}{\gamma} = \lambda e^{-\gamma d} - e^{-\gamma d}$$

$$\frac{\partial^2 U}{\partial d^2} = -\gamma\lambda e^{-\gamma d} + \gamma e^{-\gamma d}$$

We use this to determine absolute risk aversion.

$$\text{ARA} = -\frac{\frac{\partial^2 U}{\partial d^2}}{\frac{\partial U}{\partial d}} = \frac{-\gamma\lambda e^{-\gamma d} + \gamma e^{-\gamma d}}{\lambda e^{-\gamma d} - e^{-\gamma d}} = \frac{\gamma e^{-\gamma d}(\lambda-1)}{e^{-\gamma d}(\lambda-1)} = \gamma$$

$$\rightarrow \lim_{d\rightarrow\infty}\gamma = \gamma \rightarrow \text{CARA}$$

B Appendix B: Market Research Graphs