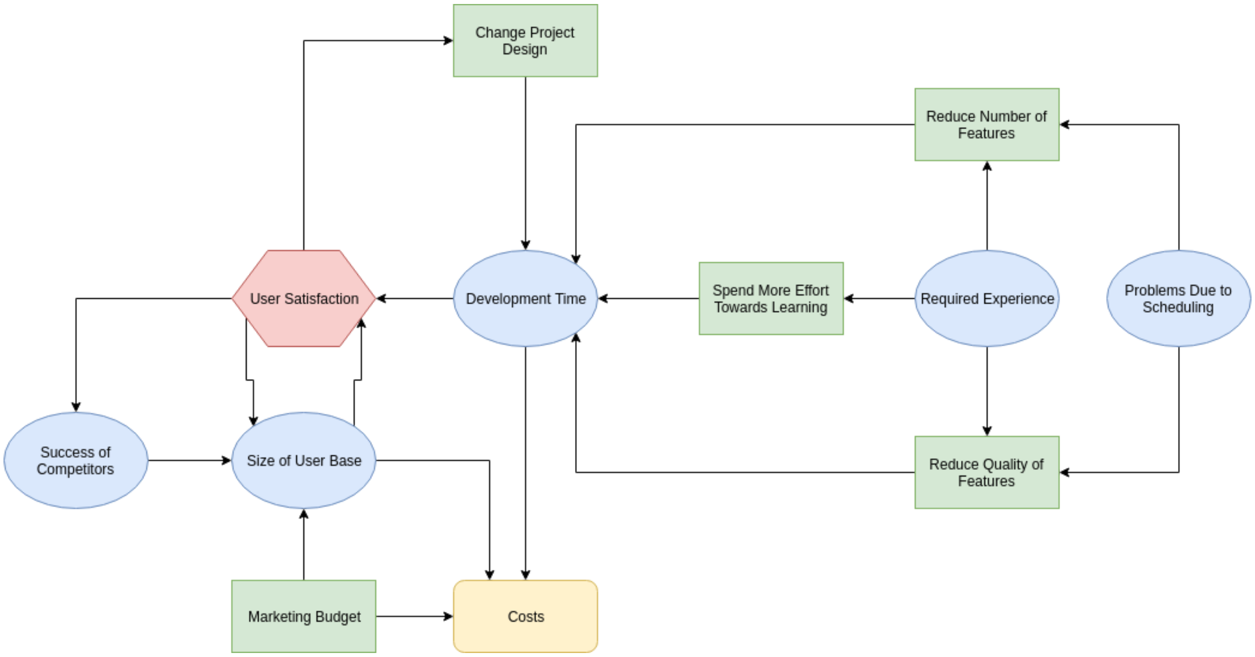
Risk Analysis

By Benson Gao, Nick McDaniel,
Michael Ly, and Julian Tran

Influence Diagram

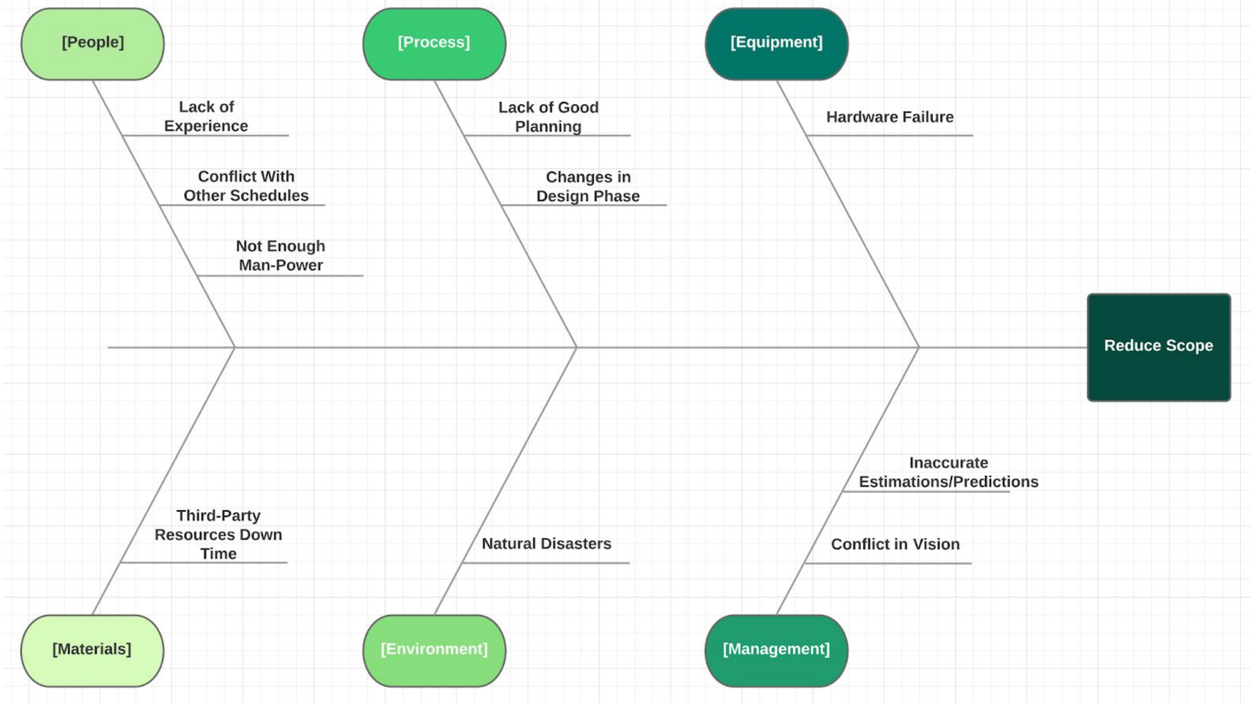


The influence diagram outlines some of the decisions that may need to be made throughout the course of the project as well as the effects they will have. The green rectangles are decisions we must make. Blue ovals are uncertainty nodes. Yellow rounded rectangles are deterministic nodes. The red hexagon is the value that we are trying to evaluate.

The right side of the influence diagram focuses more on the pre-release decisions and risks involved with our project. Risks with human resources lead into decisions about reduction of quality of features, reduction in scope, or an increase in time/effort. Regardless of the decision made, each of these affect the development time of the project. Development time directly influences the costs involved with the project as well as the objective of user satisfaction.

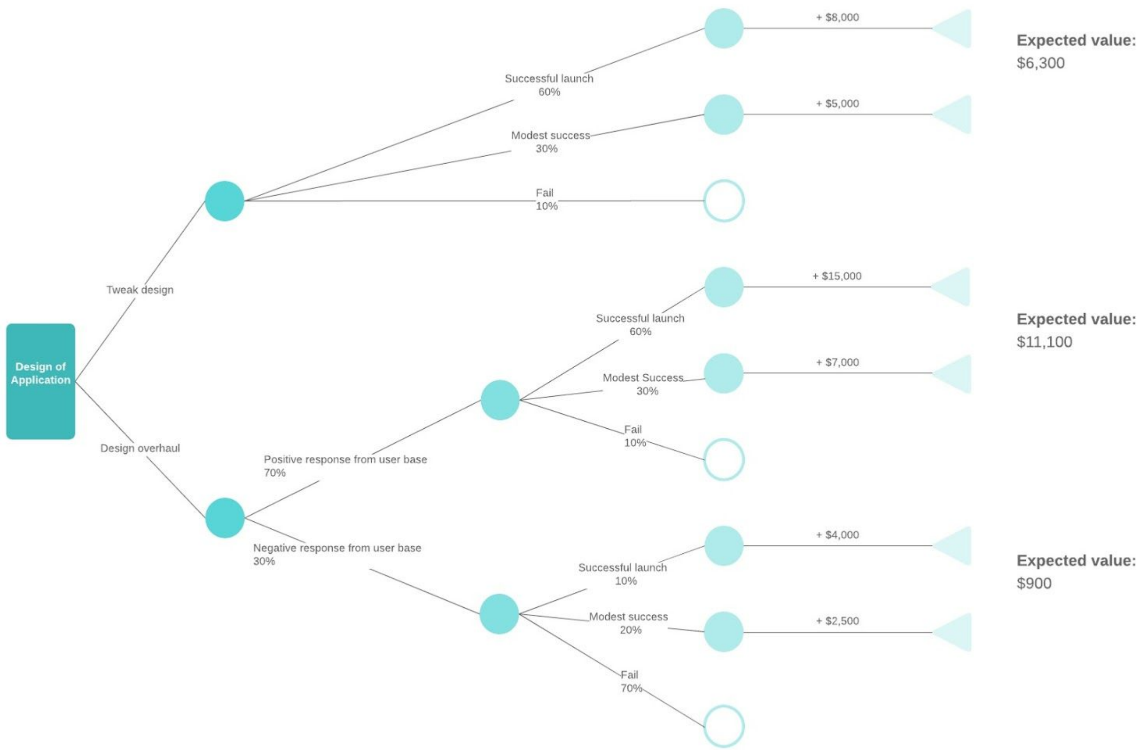
The left side of the influence diagram focuses more on the post-release development cycles involved with the project. Because we are aiming to release software that is scalable and improvable, based upon the user satisfaction, there are decisions to be made on how to change the project design to improve user satisfaction. A key aspect of our system is the size of our user base which is affected by how much we would choose to spend on marketing as well as the success of our competitors. The size of our user base is directly correlated with the server maintenance costs. User satisfaction also has a cyclical influence upon the user base. If people are satisfied with the system, the user base will grow due to word of mouth. When the user base grows, the likelihood of user satisfaction increases with more guides available at all times to help tourists. The user satisfaction also affects our competitor’s success in a similar cyclical fashion.

Ishikawa Diagram



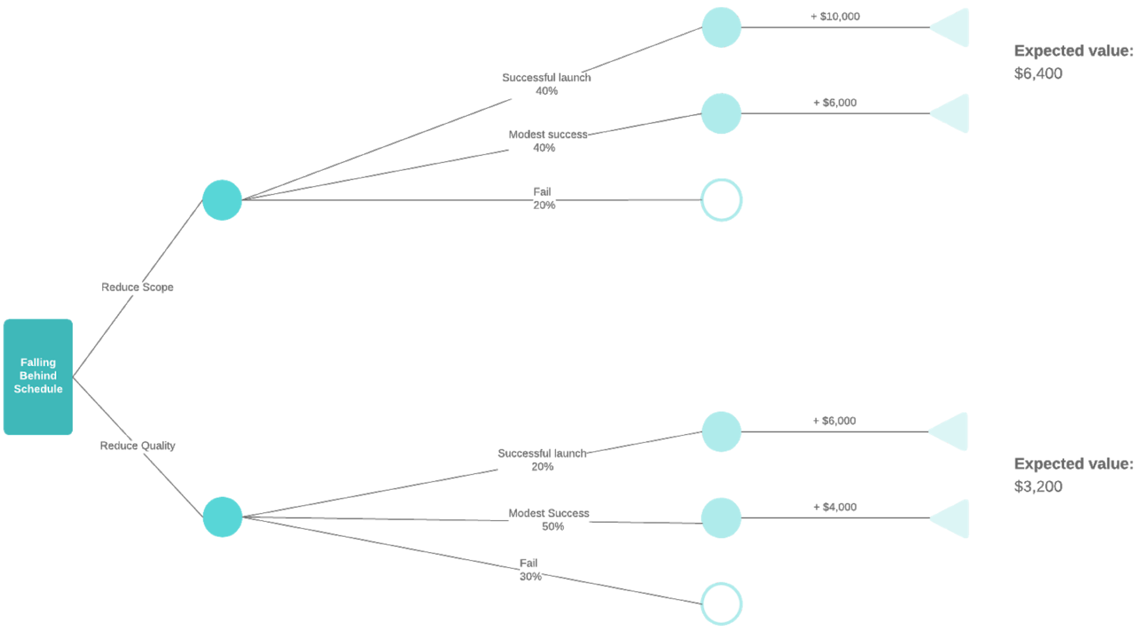
The Ishikawa diagram allows us to identify the risks that could lead to a reduced scope for our project. The sources with the largest risks are People, Process, and Management. The risks involved with our human resources are Lack of Experience, Conflict With Other Schedules, and Not Enough Man-Power. Process risks are problems with the planning and design of our project. Management risks deal with conflicts in vision and design as well as any inaccuracies with our estimations and predictions throughout the development cycle. The other three sources that may lead to a reduction in scope would be Equipment, Materials, and Environment. These three are mainly outside of our team's ability to control: Hardware Failure, Natural Disasters, and Third-Party Resource Down Time.

Decision Tree



The first decision tree shows what would happen if we decide to do something with the design of the application. We can either tweak the design, or we can do an overhaul of the design. If we tweak the design by adding a couple of things or just changing how it looks a little (i.e. font, position, etc.), there is a 60% chance of a successful launch, 30% of a modest success, and 10% of fail. The reason for the high percentages of success is reasonable because we will only be making a couple of tweaks here and there, which in return should benefit the user base.

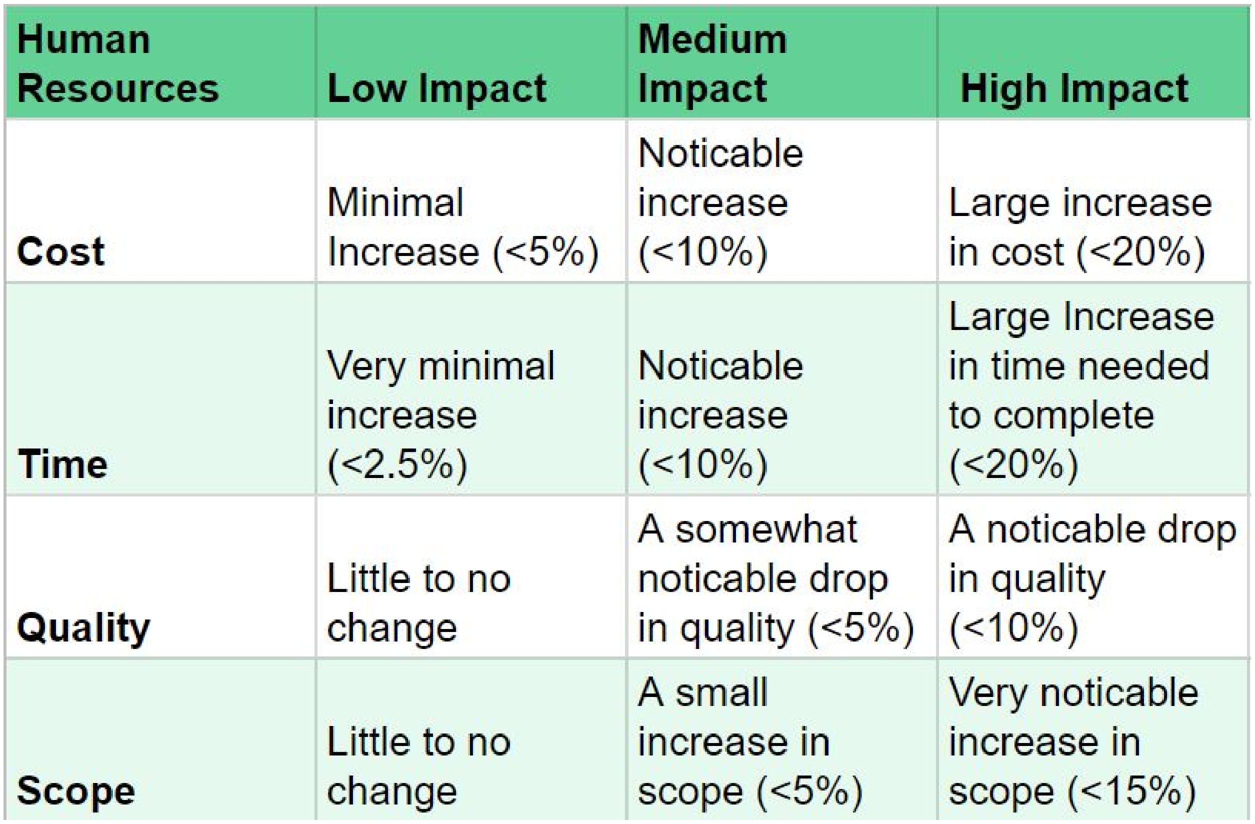
The other route that is shown is doing a design overhaul. The point of this is to change the whole layout and format of the application, possibly to keep it modern and satisfactory enough for the user base. There is a 70% chance that the overhaul is positive, and 30% that it is negative. The reason for this is the overhaul is intended to keep the application up to date, and having a nice design for users to keep using it or invite new users. There is a chance though that current users or new users may not be welcoming to the new design, which is the negative response. If it becomes a positive response, that leads into possibilities of whether the launch is successful, moderately successful, or a fail. A positive response would be more likely to lead to a successful launch, and vice versa.



The second decision tree is about mitigating the impact when our risks occur. The main decision is whether or not to reduce the scope or reduce the quality of the application when the project falls behind schedule. We determined that reducing scope but releasing an overall higher quality product would be more likely to result in a successful launch. Even in the case of a successful launch through reducing quality, it is unlikely to compare favorably to a successful launch through reducing the scope.

List of Possible Risks:

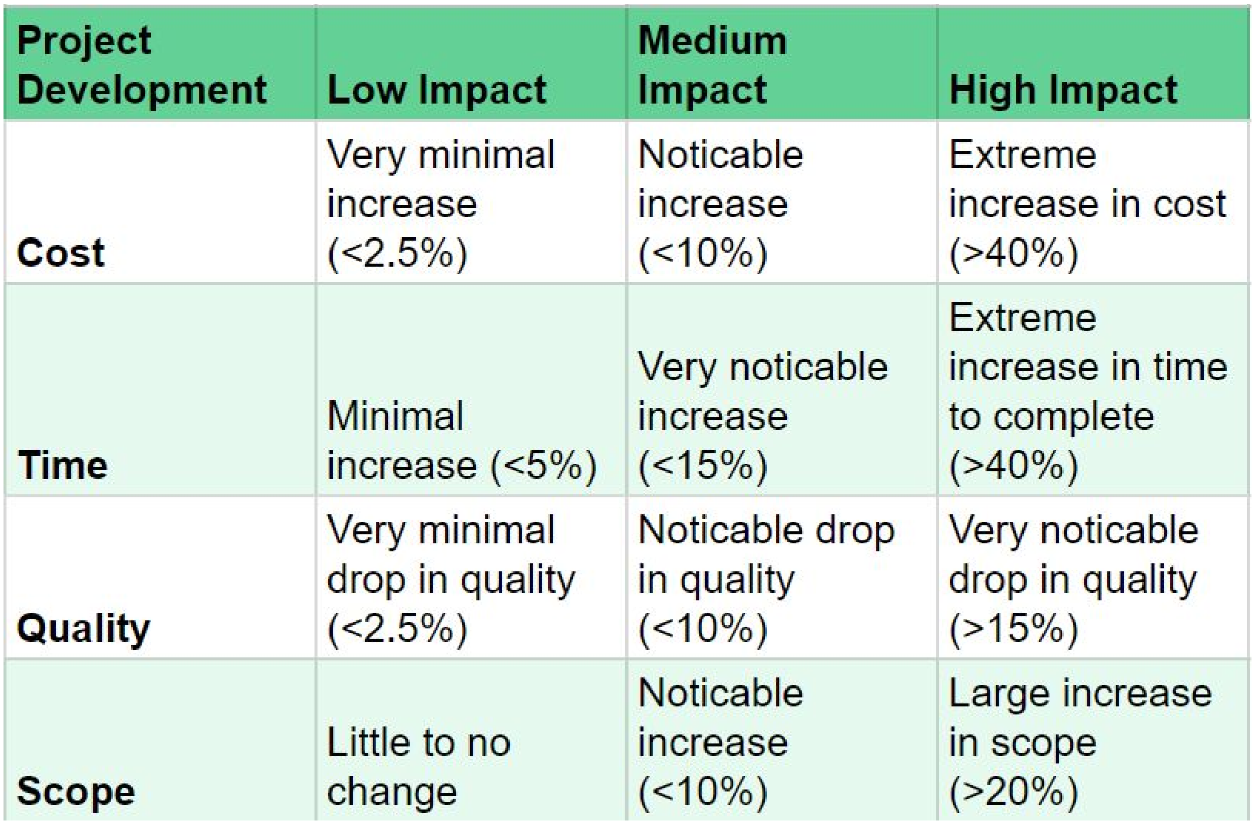
- Lack of prior experience in Android development and Firebase
    - Probability of Risk: High
    - Impact: Medium
    - Potential Loss: Delays in milestones and project delivery due to extra time needed to familiarize engineers with the technologies involved. More complicated issues that are difficult to solve may arise due to inexperience. Other issues may arise due to an improper fit between the technologies chosen and the requirements of the project.
    - Countermeasure: Spending time over in the down-time from December to January learning basics of the technologies involved.
- Other priorities and scheduling differences due to work/classes
    - Probability of Risk: Medium-High
    - Impact: Medium
    - Potential Loss: There will be a lack of communication between members, and may be unclear on who’s working on what. Some members may have to do more work than others to help make up.
    - Countermeasure: May have to spend time on a collaborative platform (such as discord + github) to work on the project together instead of meeting up in person. If possible, may have to schedule times to meet together on the weekends, which may be inconvenient to others.
- Not enough man-power with limited team (4 members)
    - Probability of Risk: High
    - Impact: Medium
    - Potential Loss: Delays in milestones and project delivery due to extra time needed. Personal emergencies would have a larger impact on the group productivity (25% of man-power compared to 20%/17% in a 5-6 man group).
    - Countermeasure: Establishing the scope of the project to be realistic for a reduced group size. Proper communication allows for other group members to help pull weight when necessary.



Human Resources	Low Impact	Medium Impact	High Impact
Cost	Minimal Increase (<5%)	Noticable increase (<10%)	Large increase in cost (<20%)
Time	Very minimal increase (<2.5%)	Noticable increase (<10%)	Large Increase in time needed to complete (<20%)
Quality	Little to no change	A somewhat noticable drop in quality (<5%)	A noticable drop in quality (<10%)
Scope	Little to no change	A small increase in scope (<5%)	Very noticable increase in scope (<15%)

- Differences between small-scale testing environments and live environment
  - Probability of Risk: Low
  - Impact: Medium
  - Potential Loss: Loss of time in transitioning between test and live application.
  - Countermeasure: Schedule development and implementation simultaneously such that our development phase isn't hindered by the implementation phase.
- Conflict in vision of the end result of the project
  - Probability of Risk: Medium
  - Impact: Medium
  - Potential Loss: Delays in the development cycle as more time must be spent reiterating the goals of the project instead of already knowing the goals and working towards them.
  - Countermeasure: Creating detailed and unambiguous diagrams that model the architecture and function of our application allow each member to have resources to check to make sure that what they are producing is progressing towards the desired end product. Weekly meetings stating what has been achieved in the last week and what should be achieved in the next week will also work to make sure that each member is aware of what they should be working on at that point of time.

- Hardware failure
    - Probability of Risk: Low
    - Impact: High
    - Potential Loss: Loss of valuable data or corruption of files that can hinder the development process. May also limit a team member's ability to contribute to development for an unspecified amount of time.
    - Countermeasure: Having the project code be placed on a Github repository allows for a backup of the main project as well as version control. It also lets a team member easily join back on the development team if they need to fix/replace hardware and are unable to work for an extended period of time.
- Inaccurate estimations/predictions
    - Probability of Risk: Low
    - Impact: Medium-High
    - Potential Loss: If the scope of the project is larger than predicted or certain requirements are more difficult to complete, then the final project may not be finished completely by the deadline. Certain components of the project may need to be cut.
    - Countermeasure: Will have to adapt, and may have to work harder to get other milestones done to finish in time. Seeking advice on our project plan while in the design phase will let us have greater foresight on our project.
- Changes in the design phase
    - Probability of Risk: Medium
    - Impact: High
    - Potential Loss: Changes in the design of the project will require reiterating the design, plan, time, and scope of the project.
    - Countermeasure: Designing all the features of the project before reaching implementation phase of the project. Creating the project such that it is possible to changes without a complete overhaul.



Project Development	Low Impact	Medium Impact	High Impact
Cost	Very minimal increase (<2.5%)	Noticable increase (<10%)	Extreme increase in cost (>40%)
Time	Minimal increase (<5%)	Very noticable increase (<15%)	Extreme increase in time to complete (>40%)
Quality	Very minimal drop in quality (<2.5%)	Noticable drop in quality (<10%)	Very noticable drop in quality (>15%)
Scope	Little to no change	Noticable increase (<10%)	Large increase in scope (>20%)

Post Project Completion Risk Analysis

Many of our risks with high probability ended up affecting the scope of the project. In the Ishikawa Diagram underneath the people category we significantly felt the impact of lack of experience with Android Studio and Firebase, conflicts in schedules, and lack of manpower. We didn't follow our own preventative measures to combat the lack of experience, we failed to put significant effort into learning Android Studio and Firebase over the break between semesters. An oversight on our part was not realizing that we were going to have different schedules next semester. Having our class start at 5pm significantly impacted our available since many activities/plans/events generally occur around 5pm. Another high risk we faced was inaccurate estimations and planning. We failed to consider other class projects and midterms into our schedule. During the midterms week, our group was unable to work outside of class since many of us were preoccupied with studying and taking exams. Having an extra person in our group we may have been able to complete the project without compromising on the scope of the project. In the end we were able to bring the proof of concept of our project by focusing on the core features and removing the supplementary features of our project. If we had more time the supplementary features can be added to enhance the user's experience. The functionality of our application is not impeded by the removal of these supplementary features.