Team Name: Moniker to be used in all presentations and documentation.

Wanderlist

Members: List of team members, first & last name.

Daniel Green, Gayathri Gude, Darian Valdez, Gordon Gu, Joanne Liu, Zack Toelkes

Description: A short (2-3 paragraphs) description of the project. Provide enough information to explain what value your product will provide to users of your product.

The project will be a web application that will be used to help users find vacation services, activities, and amenities. Users will be shown a collage of images for a randomized vacation spot in Boulder. They can either like or reject said vacation spot. Once the user has accumulated five liked vacation spots, or has gone through fifty vacation spots, their liked spots will be shown in list format with some additional information. The user can then use this information to jump start vacation planning.

By using this application, users will be able to consolidate and quickly find information for their desired activities. By only showing a picture of the location or activity it forces the user to make a decision that isn't influenced by pricing or other factors. Once the list is compiled, the user will be given information about price, the amount of time, location, and other details regarding the item.

Vision Statement: A simple, one-sentence statement describing the clear and inspirational desired state resulting from your team's efforts to create your product.

To establish a connection to vacation/tourist spots according to the current location, along with a price range.

Motivation: Describe the background and reasons for developing this product.

Finding a spot to vacation or go out can become near impossible with the obstacle of overchoice. Overchoice is where there are so many options to pick from, that no decision is made, leading to no options being selected. With the near infinite spots to vacation worldwide, overchoice becomes a problem. Our app limits user choice, but still allows user input to shape the vacation selection process. This will spur people to choose a spot to vacation, instead of simply becoming overwhelmed by all the possibilities.

Risks: What are the known risks that may prevent your team from completing this project on time. Risks could include: the working environment, lack of experience of the team in the area of focus, lack of access to a specific resource, etc.

So far, the team has limited weakly hours to work on the project. To accomplish all tasks with the given time frame will challenge the team's efficiency. However, as far as experience and

access to resources, we have a broad range of skills that all members are able to offer, along with many resources, in which will serve as a strength.

Risk Mitigation Plan: A detailed plan showing how the team will mitigate each stated risk. Describe how you will succeed given the stated risks.

Weekly full team meetings will be used to check status of project and keep the team on track for finishing the project by semester's end. Each meeting will last a duration of an hour, within which we will discuss action items of the past week, and set out action items for the following week and beyond.

Version Control: Describe the version control method and repository you will be using for the deliverables created for the project. Github is strongly recommended. Once a repository is determined, you must share access to the repository with your instructor, your TA, and all your project team members.

GitHub will act as our repository and version control. Moving to a new version of the developed software will be handled in one of two ways. High level changes, such as software structure and feature addition, will need to be reviewed by the team before being commited. Simple debugging can be committed without full team review, if another member has reviewed it, a detailed message explaining what was changed is attached and the effected code is bracketed and changes commented.

Development Method: Which software development methodology will your team follow? Describe the methodology and the features/steps you will follow. Common methodologies include waterfall, agile/scrum, iterative, spiral, etc.

The team will continue to follow an agile development methodology. Due to the team's inexperience and the limited time frame, the project's scope will vary as the team finds out the speed at which components of the software can be developed. In addition, an agile methodology allows for team members to reorganize sub teams as the project progresses. This may become necessary during development of the front end, back end, and server side software, since members may find they excel in one region over the others. Following an agile method, we started out by selecting a project idea from a brainstorming session. We scoped out features, necessary resources, and technology to be utilized throughout the semester. Within the next couple of weeks, we will be exploring tools and templates that will be used for our project. We will then plan the project, such that it abides by our skill set, time frame, and budget.

Collaboration Tool: Select a collaboration tool for team members to utilize for coordination of their work and communication among team members. Popular tools are Slack and HipChat.

We are currently using Facebook Messenger to update the team on action items and where meetings are being conducted. In the coming weeks we may transition to Slack if we required more structured communication. We will also be using google drive as a means to archive our thoughts, in a way that we can revisit throughout our process.

Proposed Architecture: Propose an architecture for your app. What technologies will you be using on the backend? What technologies on the front end? How will they communicate with each other? Which technologies will be responsible for which functionalities?

Our project will be a web application built with the ASP.NET framework with Visual Studio. The application will be driven by a C# logic layer, and the frontend will be HTML (via Adam Live coding). The database backend will be driven by SQLite because it is serverless and lightweight, with zero configuration.