Omar AlSughayer (oasugher@cs)
Jessica Bahr(jbahr99@cs)

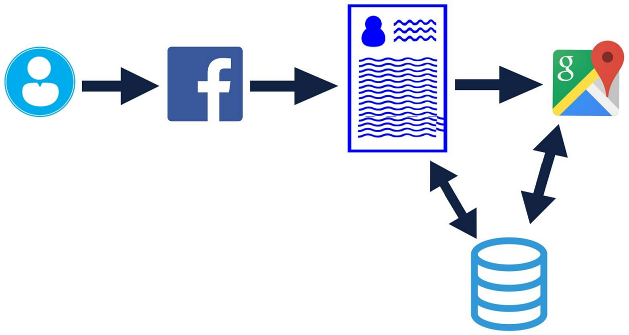
GeoDare

Vision and Novelty

GeoDare is a real-world app that allows users to create or attempt challenges present in their current geographical location. It is the perfect way to explore a new city or enjoy your own city away from usual touristy activities. The nature of its challenges and social networking sets it aside from other real-world games that are centered around finding hidden objects, such as GeoCache and Pokemon Go, which require the user to travel and hunt for items at a specific location. Our app has the user not only travel to a specific location but also perform challenges at that location. GeoDare is also different from other party challenge games, such as 7 Second Challenges, since they require participants to perform challenges within the confines of their home in a set time limit. Our app makes the user travel around a city performing challenges that do not have a time limit. People who use this app will have the satisfaction of completing crazy challenges outside their home while at the same time being able to explore a city in a unique way. An example of these challenges in the UW campus would be to climb a cherry blossom tree in the Quad or jump into Drumheller Fountain. Our app proposes to solve the problem of tourists and residents who want to see their city from a unique perspective through the usage of providing challenges to be completed while within the city. We believe that there is an increasing trend of bringing games and other activities to the real-world and this app would be a part of that trend. Since users are the ones creating the challenges within the app, there will always be new, unique challenges available every day.

Architecture

The GeoDare Software Architecture includes multiple components that will need to be integrated together in order for the app to work. The first component is to use Facebook Login as the security and authentication behind the app. The user will login to the app using their Facebook account then their GeoDare profile will show up. This is indicated in the below diagram with the Facebook symbol and the profile page. Every user's profile will host their achievements, completed and created challenges, and the ability to attach videos or images to any challenge as proof of completion. The location aspect of the app will be created using the Google Maps' framework. Using it, the user will be able to view nearby challenges and create their own challenges from their current location as denoted by the Google Maps icon in the below diagram. All the data about the user profiles will be stored in a SQL database for easy storage and searching capabilities as denoted by the database icon in the below diagram. From a technical standpoint, the most interesting part is to figure out how to integrate all of these components together into an interactive and easy to use GUI. We are planning on using Java as our main programming language with Android SDK as the main library within Java. We will expand further by using frameworks such as Facebook SDK, Google Maps API, and MySQL databases.



Challenges and Risks

The most challenging aspect of this app will be to integrate all of the libraries, frameworks, and database information into a working form. Working across multiple resources always poses a risk of not being able to integrate them correctly. For example, trying to coordinate Google Maps interface to be used within the GUI of the app. The best way we have figured out to minimize this risk is to designate a team member as an integrator whose purpose is to help integrate all the components together either through interfaces or data transfers. Along with this, there is the problem of being able to find a team that is well-balanced and contains people with different levels of expertise working with each library/framework. If we cannot create a well-balanced team, then another challenge is to make sure that we provide enough time within our development process to allow team members to learn and work with these libraries/frameworks while still finishing the app on time. For example, if we do not have anyone on the team that has experience with the Facebook SDK, then someone within the team will need the time to learn the library. The best way we have figured out to minimize this risk is to overestimate on the time needed to educate the entire team. If we have extra time due to overestimation, then we can use that time to provide more features to the app.